# Statistics for Data Science, Homework #1

Sofia Pietrini, 258112

02/04/2025

## Introduction

The data analyzed in this homework comes from a cardiovascular study conducted in the US, focused on investigating the possible risk factors associated to the development of coronary heart disease within 10 years. In particular, the risk factors available in the data are: the patients' sex (`sex`), age (`age`), education level (`education`, coded as 1: no high school degree, 2: high school graduate, 3: college graduate, 4: post-college), current smoking status (`smoker`), number of cigarettes per day (`cpd`, continuous), previous occurrence of strokes (`stroke`) or hypertension (`HTN`), presence of diabetes (`diabetes`), cholesterol levels (`chol`, continuous), diastolic blood pressure (`DBP`, continuous), body mass index (`BMI`, continuous), and heart rate (`HR`, continuous). The variable `CHD` records whether the patient developed CHD in the next 10 years or not.

## Loading data set, data exploration and pre-processing

The libraries needed for the study are: `tidyverse`, `tidymodels`, `ISLR2`, `class`, and `caret`.

```
dataf <- read.csv("chd.csv")
dim(dataf)
```

```
## [1] 4238   13
```

Since the data set is composed of 4238 observations, it is useful to view it as a tibble, which truncates automatically the output and can give us insights on the datatype of each column.

```
as_tibble(dataf)
```

```
## # A tibble: 4,238 x 13
##    sex      age education smoker   cpd stroke   HTN diabetes  chol   DBP   BMI
##    <chr>  <int>     <int>  <int> <int>  <int> <int>    <int> <int> <dbl> <dbl>
##  1 Male      39         4      0     0      0     0        0   195    70  27.0
##  2 Female    46         2      0     0      0     0        0   250    81  28.7
##  3 Male      48         1      1    20      0     0        0   245    80  25.3
##  4 Female    61         3      1    30      0     1        0   225    95  28.6
##  5 Female    46         3      1    23      0     0        0   285    84  23.1
##  6 Female    43         2      0     0      0     1        0   228   110  30.3
##  7 Female    63         1      0     0      0     0        0   205    71  33.1
##  8 Female    45         2      1    20      0     0        0   313    71  21.7
##  9 Male      52         1      0     0      0     1        0   260    89  26.4
## 10 Male      43         1      1    30      0     1        0   225   107  23.6
## # i 4,228 more rows
## # i 2 more variables: HR <int>, CHD <chr>
```

It is clearly visible how the two non-numerical variables, `CHD` and `sex`, are char variables, whereas the numerical ones are almost all integer, except `DBP` and `BMI` which are double variables. The response variable (`CHD`) is a categorical variable that can take only two labels: *"Yes"* if the patient developed CHD in the past 10 years, *"No"* otherwise. Consequently, `CHD` was transformed into a factor, as well as the other char variable `sex`, with the

command `dataf$variable <- factor(dataf$variable)`.
Another key feature of the data set is the prevalence of binary variables (`smoker`, `stroke`, `HTN` and `diabetes`) and one ordinal variable (`education`). These were, also, transformed into factors to facilitate the analysis.
These transformations help us to have a better understanding of the data we are working with as well as the overall characteristics of the patients. It, also, highlighted the imbalance of the data set with respect to the response variable `CHD`, since we can see from the summary that, out of the 4238 observations, 3594 (85%) did not develop CHD in the next ten years, while 644 (15%) did.

```
summary(dataf)
```

```
##      sex            age        education   smoker        cpd         stroke
##  Female:2419   Min.   :32.00   1   :1720   0:2144   Min.   : 0.000   0:4213
##  Male  :1819   1st Qu.:42.00   2   :1253   1:2094   1st Qu.: 0.000   1:  25
##                Median :49.00   3   : 687            Median : 0.000
##                Mean   :49.58   4   : 473            Mean   : 9.003
##                3rd Qu.:56.00   NA's: 105            3rd Qu.:20.000
##                Max.   :70.00                        Max.   :70.000
##                                                     NA's   :29
##  HTN      diabetes      chol            DBP              BMI
##  0:2922   0:4129   Min.   :107.0   Min.   : 48.00   Min.   :15.54
##  1:1316   1: 109   1st Qu.:206.0   1st Qu.: 75.00   1st Qu.:23.07
##                    Median :234.0   Median : 82.00   Median :25.40
##                    Mean   :236.7   Mean   : 82.89   Mean   :25.80
##                    3rd Qu.:263.0   3rd Qu.: 89.88   3rd Qu.:28.04
##                    Max.   :696.0   Max.   :142.50   Max.   :56.80
##                    NA's   :50                       NA's   :19
##        HR          CHD
##  Min.   : 44.00   No :3594
##  1st Qu.: 68.00   Yes: 644
##  Median : 75.00
##  Mean   : 75.88
##  3rd Qu.: 83.00
##  Max.   :143.00
##  NA's   :1
```

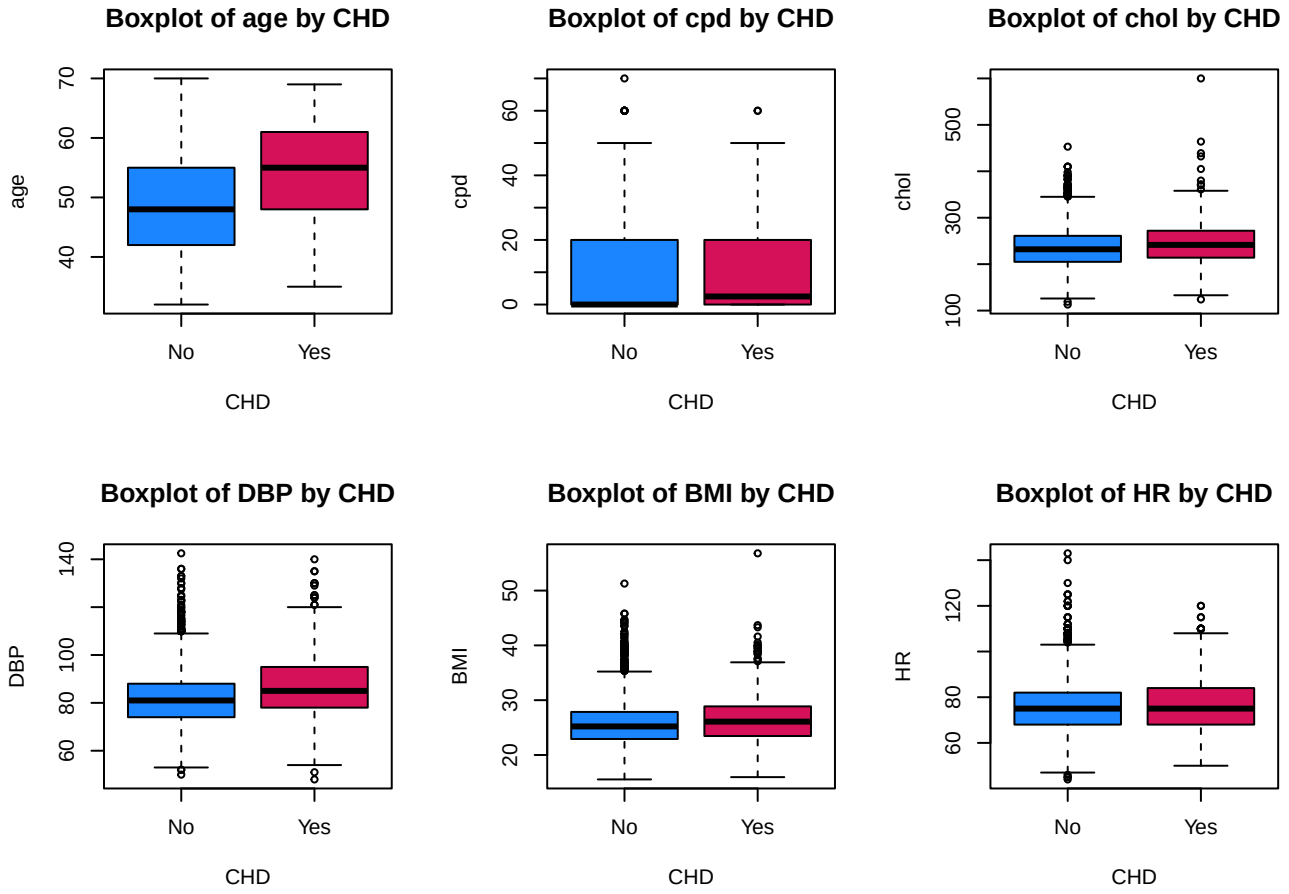
Another takeaway from the summary is the presence of NA's. Given the presence of so many binary and ordinal variables and the wish to give coherence to the method, we choose to omit all of them, instead of substituting those present in the continuous predictors' columns with the corresponding variable mean.

```
dataf <- na.omit(dataf)
dim(dataf)
```

```
## [1] 4039   13
```

By omitting all the NA's, we loose 199 observations, corresponding to 4.6% of the data set.

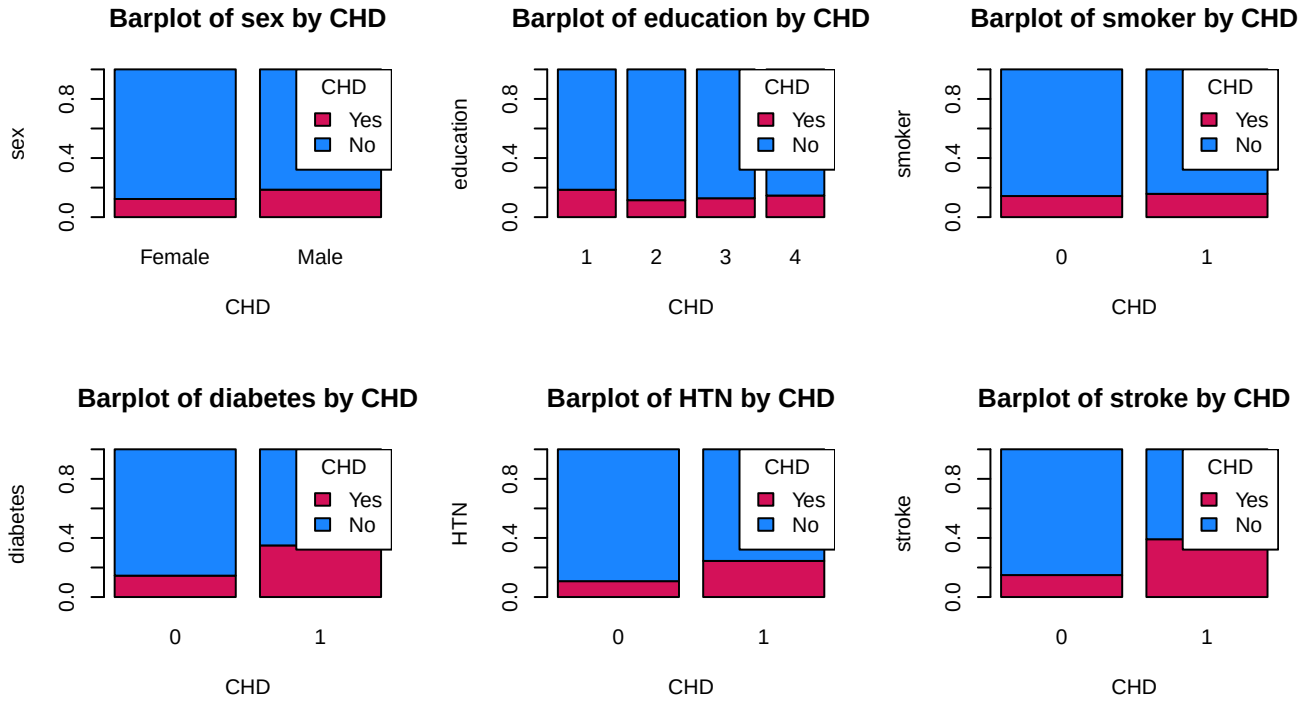
```
# Visualization of the discriminative power of continuous predictors
predictors <- c("age", "cpd", "chol", "DBP", "BMI", "HR")
response <- "CHD"
colors <- c("#1A85FF", "#D41159")
# Create a function to plot boxplots for all continuous predictors
plot_boxplots <- function(data, predictors, response) {
    par(mfrow = c(2, 3))  # Arrange plots in a 2x3 grid
    for (predictor in predictors) {
        boxplot(data[[predictor]] ~ data[[response]], main = paste("Boxplot of",
            predictor, "by", response), xlab = response, ylab = predictor, col = colors)
    }
    par(mfrow = c(1, 1))  # Reset layout to default
}
plot_boxplots(dataf, predictors, response)
```




The boxplots reveal both the distribution of continuous variables and their discriminative power. `age`, `DBP`, and `BMI` emerge as potentially significant predictors, with `chol` showing weaker association.
For categorical variables, proportional barplots better illustrate relationships with `CHD`.

```
# Define categorical variables
predictors <- c("sex", "education", "smoker", "diabetes", "HTN", "stroke")
response <- "CHD"
colors <- c("#D41159", "#1A85FF")

# Create a function to plot barplots for all categorical predictors
plot_barplots <- function(data, predictors, response) {
    par(mfrow = c(2, 3))  # Arrange plots in a 2x3 grid
    for (predictor in predictors) {
        freq_table <- table(data[[predictor]], data[[response]])
        freq_table <- freq_table[, rev(colnames(freq_table))]  #invert the order of the columns
        prop_table <- prop.table(freq_table, margin = 1)
        # convert into proportion, to scale the columns
        barplot(t(prop_table), width = 1, main = paste("Barplot of", predictor, "by",
            response), xlab = response, ylab = predictor, col = colors)
        legend("topright", legend = rev(levels(as.factor(data[[response]]))), fill = colors,
            title = response, bg = "white")
    }
    par(mfrow = c(1, 1))  # Reset layout to default
}
plot_barplots(dataf, predictors, response)
```




All the categorical predictors, with the exception of `smoker`, seem to have a certain degree of discriminative power.

## Split the data into training and test sets

As previously illustrated, the given data set present a slightly imbalance with respect to the response variable `CHD`, with the label "Yes" being observed only in 15% of the total observations. Given the class imbalance, the aim in splitting the data into training and test set was to maintain the same imbalance with respect to the outcome variable. For this purpose, we used the dedicated function `createDataPartition(labels, p=train_size)` from the library `caret`. The function was tested using different values (0.5, 0.7, 0.75 and 0.8) for the parameter p, which is a number between 0 and 1 representing the percentage of data used for training. All the values listed before resulted in the same, or significantly similar, class imbalance, so the choice made was to do a 70-30 split.

```
set.seed(1)  # For reproducibility
# Get a vector of length 0.7*n of random indices
test <- createDataPartition(dataf$CHD, p = 0.7, list = FALSE)
# Split data into training and test set
dataf_tr <- dataf[-test, ]
dataf_ts <- dataf[test, ]
# Get CHD for test set
CHD_test <- dataf_ts$CHD  #Returns a factor with 2 levels
# Transform the factor data into labels for future operations
CHD_test <- as.character(CHD_test)

# Check the distribution of CHD in training and test sets
prop.table(table(dataf_tr$CHD))
```

```
##
##        No       Yes
## 0.8504132 0.1495868
```

```
prop.table(table(dataf_ts$CHD))
```

```
##
```

```
##        No       Yes
## 0.8497702 0.1502298
```

# Fit GLM model

In this section we are going to fit the following GLM model:

$$\text{logit}(\text{E}(\text{CHD})) = \beta_0 + \beta_1 \text{sex} + \beta_2 \text{age} + \beta_3 \text{education} + \ldots + \beta_{12} \text{HR}$$

The model is logistic regression, which estimates the log-odds of developing CHD based on the predictors available in our data set.
Each coefficient $\beta_i$ will represent the log-odds change associated with a one-unit increase in that predictor while holding all other variables constant.

```
glm.fit <- glm(CHD ~ sex + age + education + smoker + cpd + HTN + diabetes + chol +
    DBP + BMI + HR + stroke, data = dataf_tr, family = binomial)
# family=binomial selects a Logistic Regression model
summary(glm.fit)  #visualize output of the logistic regression model
```

```
##
## Call:
## glm(formula = CHD ~ sex + age + education + smoker + cpd + HTN +
##     diabetes + chol + DBP + BMI + HR + stroke, family = binomial,
##     data = dataf_tr)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.0085719  1.1912293  -5.883 4.02e-09 ***
## sexMale      0.4549143  0.1899834   2.394   0.0166 *
## age          0.0717425  0.0112818   6.359 2.03e-10 ***
## education2  -0.2443417  0.2161335  -1.131   0.2583
## education3  -0.1917152  0.2559041  -0.749   0.4538
## education4  -0.1121488  0.2889175  -0.388   0.6979
## smoker1     -0.2175265  0.2808255  -0.775   0.4386
## cpd          0.0264474  0.0105739   2.501   0.0124 *
## HTN1         0.4772300  0.2205620   2.164   0.0305 *
## diabetes1    0.5756021  0.4063908   1.416   0.1567
## chol         0.0009613  0.0020062   0.479   0.6318
## DBP          0.0085846  0.0084368   1.018   0.3089
## BMI         -0.0195921  0.0232048  -0.844   0.3985
## HR           0.0080174  0.0076368   1.050   0.2938
## stroke1      1.8634125  0.8496286   2.193   0.0283 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1021.22  on 1209  degrees of freedom
## Residual deviance:  913.17  on 1195  degrees of freedom
## AIC: 943.17
##
## Number of Fisher Scoring iterations: 5
```

Among all the information given by the summary, for our study we are going to focus on the estimates of our coefficients and their p-values, hence their significance. As a confirmation to what discussed before, the variables `sex`, `age`, `HTN` and `stroke` exhibit a low p-value. A confirmation to the fact that these variables have an higher discriminative power as predictors for CHD.
Contrary to initial exploratory analysis, the GLM identified `cpd` (cigarettes per day) as statistically significant

(p=0.0157). Notably, variables like `diabetes` and `chol`, while suggestive in visualizations, lacked significance in the model. However, proceeding with the summary, the model's overall fit is supported by the reduction in deviance from the null model (1021.22) to the residual deviance (917.51), indicating that including all predictors improves the model's explanatory power, even though not all variables contribute equally to predicting CHD risk.

Let's now compute the accuracy of the model, by getting prediction out of it, with the `predict()` function called over the test set. Then, we can build our confusion matrix (rows=truth, columns=predictions) and compute the number of correct predictions / total observations (the accuracy), or the prediction error.

```
glm.probs <- predict(glm.fit, data = dataf_ts, type = "response")
# type='response' returns predictions in term of probabilities

# glm.probs presents the probability of CHD being 'Yes' since No=0 and Yes=1
contrasts(dataf$CHD)
```

```
##     Yes
## No    0
## Yes   1
```

```
# create a 'placeholder' filled with as many No values as the number of
# observations (rows) in dataf
glm.pred <- rep("No", nrow(dataf_ts))
# replace with Yes values according to the glm.probs threshold, by default is
# 0.5
glm.pred[glm.probs > 0.5] <- "Yes"
# | tidy: false in this case Yes will be the positive value
table(glm.pred, CHD_test)
```

```
##         CHD_test
## glm.pred   No  Yes
##      No  2384  416
##      Yes   20    9
```

```
mean(glm.pred == CHD_test)  #accuracy = number of correct hits/total observations
```

```
## [1] 0.8458819
```

```
mean(glm.pred != CHD_test)  #error rate
```

```
## [1] 0.1541181
```

The accuracy of the model is 84%, which, in theory, is a good value. This model could still be improved, since it produces a lot of false negatives and the accuracy in predicting "Yes" is very low.

## Fit K-NN classifier

In this section we are going to investigate if the performance of the model improves when using a different approach: a K-Nearest Neighbor (KNN) model.
In R, the KNN is available through the `knn(train, test, factor_of_true_classification, number_neighbours)` function in the `class` library. However, opposite from `glm()`, you do not fit and predict, but you get the predictions using a single command: `knn()`. Firstly, we have to do some pre-processing of the data sets we are going to use. KNN works only with continuous variables, so we are going to do a selection of the predictors used for the analysis.

```
X_train <- dataf_tr %>%
    select(age, cpd, chol, DBP, BMI, HR)
X_test <- dataf_ts %>%
    select(age, cpd, chol, DBP, BMI, HR)
# Get CHD for training set
CHD_train <- dataf_tr$CHD
```

Now, we can fit our KNN model, by setting a random seed and calling the function `knn()`. The setting of the random seed is needed as a certain level of randomness is involved with KNN, e.g. when there's a tie between classes and the algorithm needs to choose one randomly.

```
set.seed(1)
knn.pred <- knn(X_train, X_test, CHD_train, k = 7)

# confusion matrix
table(knn.pred, CHD_test)
```

```
##         CHD_test
## knn.pred   No  Yes
##      No  2371  408
##      Yes   33   17
```

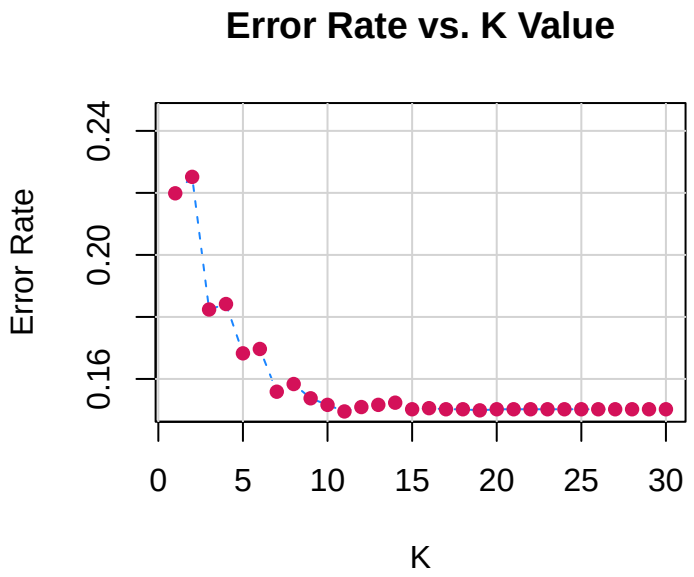
```
# accuracy
mean(knn.pred == CHD_test)
```

```
## [1] 0.8441145
```

The value k=7 was chosen based off of random trials with different values of the parameter and a more precise evaluation conducted with the elbow method, a technique with which we can select a value for k by looking at the result given by every k.
We iterate over a range of i values, here, we iterated from 1 to 30.
For each i, we calculate the error rate of the KNN method.

```
set.seed(1)
error_rate <- numeric()

for (i in 1:30) {
    knn.pred <- knn(X_train, X_test, CHD_train, k = i)
    error_rate <- c(error_rate, mean(knn.pred != CHD_test))
}
plot(1:30, error_rate, type = "l", col = "#1A85FF", lty = 2, pch = 16, main = "Error Rate vs. K Value",
    xlab = "K", ylab = "Error Rate", ylim = c(0.15, max(error_rate) + 0.02))
grid(lty = "solid")
points(1:30, error_rate, col = "#D41159", pch = 16, cex = 1)
```



Once generated the plot we can observe that the error rate tends to decrease as the number of neighbors increases, reaching the minimum for $k = 19$. Despise that, we preferred not to choose such a high value of k as the tuning parameter. In fact, as k increases, the model tends to have a high bias and a low variance, and hence can underfit the data.
To reach a good bias/variance trade-off we want to look for the value of k after which the error becomes constant, or barely decreases. In our case this would be 7.
As illustrated, the KNN model with k=7 gives us an accuracy of 84.41% which is not an improvement from the logistic regression model of the previous section.

By fitting the model with k=19, we could also note how maximum accuracy for the KNN model, ~85%, is obtained by always predicting "No", except for a single correct positive prediction. For $k \geq 21$ the model still has an accuracy of ~85%, but obtained by always predicting "No". This is a direct consequence of the imbalance of our data set with respect to the response variable `CHD`.

```
set.seed(1)
knn.pred <- knn(X_train, X_test, CHD_train, k = 21)
# confusion matrix
table(knn.pred, CHD_test)
```

```
##         CHD_test
## knn.pred   No  Yes
##      No  2404  425
##      Yes    0    0
```

```
# accuracy
mean(knn.pred == CHD_test)
```

```
## [1] 0.8497702
```

# Conclusion

In this study we analyzed how to fit two classification methods, Logistic Regression and KNN, to do prediction on variable of interest, `CHD`, evaluating their performances. The two models exhibited very similar accuracies: the majority of tests gave an accuracy of ~84%, with KNN being slightly more imprecise when tuning the parameter k=7 (carefully chosen in the previous section).
Given the imbalance of the data set with respect to the response variable, by using KNN the maximum accuracy (85%) was reached by always predicting "No".
As a direct consequence, in theory, we could consider the two models to be suitable for predicting the response variable given the predictors. However, while both models achieved similar accuracy (~84%), neither is clinically useful for identifying at-risk patients. The GLM correctly classified only 9 of 425 "Yes" cases (Sensitivity=2.1%), and KNN showed marginally better but still inadequate performance (14 TP).
Due to the nature of the question, a model suitable in correctly predicting the occurrence of this disease should be trained in a way that prediliges precision (reliability of positive predictions), since it's more important to have fewer false negatives as possible. Neither of the two models are successful in doing this, since the number of false positives is much higher than the one of true positives in both cases.
The main limitation of this study, as we could observe, was the severe class imbalance, which made accuracy misleading, as models could achieve 85% accuracy by simply predicting "No" for all cases. Another matter to take into consideration would be the loss of information derived from the omission of the observations presenting Na's (operation which could also introduce biases) and the exclusion of categorical variables when predicting information with KNN.
GLM, on the other hand, assumes linear log-odds relationships, missing potential other interaction between variables. Also, its fixed 0.5 threshold is unsuitable for imbalanced data.
Lastly, validation relied on a single 70-30 split without cross-validation, risking overfitting.
These limitations highlight the need for more sophisticated approaches to handle class imbalance and validate predictive models in future research on CHD risk factors.