Homework 1

Anthony Tricarico (254957)

2025-03-23

1 Introduction

This report will present the results from a statistical analysis whose primary goal is to find the main predictors of Coronary Heart Disease (CHD). Moreover, the relevant code to replicate the analysis will be included in the report while the complete script is available in a dedicated GitHub repository.

2 Data Cleaning and Preprocessing

It is fundamental to check the integrity of the dataset and show the count of missing values per variables.

Table 1: Summary of the Dataset

sex	age	education	smoker	cpd	stroke	HTN	diabetes	chol	DBP	BMI	HR	CHD
Length:4238	Min. :32.00	Min. :1.000	Min. :0.0000	Min. : 0.000	Min. :0.000000	Min. :0.0000	Min. :0.00000	Min. :107.0	Min. : 48.00	Min. :15.54	Min. : 44.00	Length:4238
Class :character	1st Qu.:42.00	1st Qu.:1.000	1st Qu.:0.0000	1st Qu.: 0.000	1st Qu.:0.000000	1st Qu.:0.0000	1st Qu.:0.00000	1st Qu.:206.0	1st Qu.: 75.00	1st Qu.:23.07	1st Qu.: 68.00	Class :character
Mode :character	Median :49.00	Median :2.000	Median :0.0000	Median : 0.000	Median :0.000000	Median :0.0000	Median :0.00000	Median :234.0	Median : 82.00	Median :25.40	Median : 75.00	Mode :character
NA	Mean :49.58	Mean :1.979	Mean :0.4941	Mean : 9.003	Mean :0.005899	Mean :0.3105	Mean :0.02572	Mean :236.7	Mean : 82.89	Mean :25.80	Mean : 75.88	NA
NA	3rd Qu.:56.00	3rd Qu.:3.000	3rd Qu.:1.0000	3rd Qu.:20.000	3rd Qu.:0.000000	3rd Qu.:1.0000	3rd Qu.:0.00000	3rd Qu.:263.0	3rd Qu.: 89.88	3rd Qu.:28.04	3rd Qu.: 83.00	NA
NA	Max. :70.00	Max. :4.000	Max. :1.0000	Max. :70.000	Max. :1.000000	Max. :1.0000	Max. :1.00000	Max. :696.0	Max. :142.50	Max. :56.80	Max. :143.00	NA
NA	NA	NA's :105	NA	NA's :29	NA	NA	NA	NA's :50	NA	NA's :19	NA's :1	NA

The variables `education`, `smoker`, `cpd`, `chol`, `BMI`, and `HR` all contain at least one missing value. Rows containing a missing value in one of these columns will be dropped from the dataset.

```
df <- na.omit(df)
```

It is also important to check the structure of the dataset to see if all the variables are cast to the correct type.

Using the `str` function it is possible to see that most of the variables are not cast to their correct type. For instance, all categorical variables are not coded as factors but either as numeric or character values.

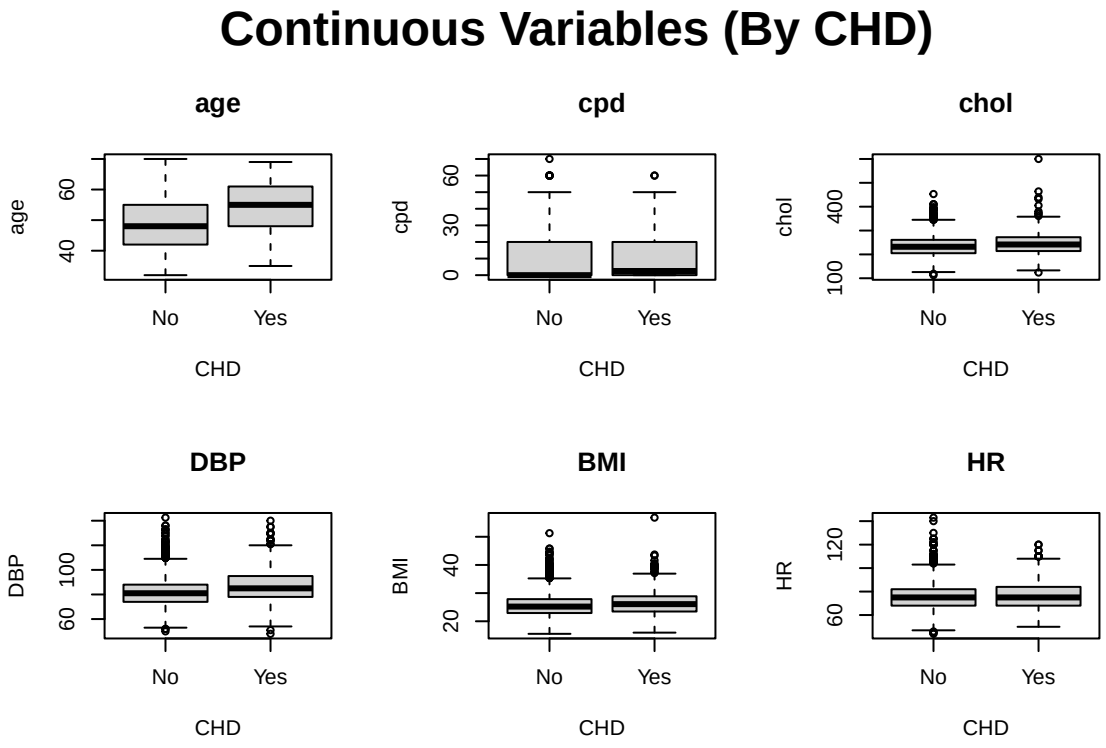
This might cause problems later on, so we address this problem by casting each variable to its correct data type.

```
df_cleaned <- df |>
  mutate(sex = as.factor(sex),
         education = factor(education, levels = c(1, 2, 3, 4),
                            labels = c('NoHS', 'HS', 'COL', 'P-COL')),
         smoker = factor(smoker, levels = c(0,1), labels = c('No', 'Yes')),
         stroke = factor(stroke, levels = c(0,1), labels = c('No', 'Yes')),
         diabetes = factor(diabetes, levels = c(0,1), labels = c('No', 'Yes')),
         HTN = factor(HTN, levels = c(0,1), labels = c('No', 'Yes')),
         CHD = as.factor(CHD))
```

Now that the data has been cleaned, we are ready to proceed to data exploration.

3 Exploratory Data Analysis

The name of categorical and numerical variables have been stored in two different vectors, namely `categ` and `numer`. This was done to conveniently plot data based on whether they were categorical or numerical.



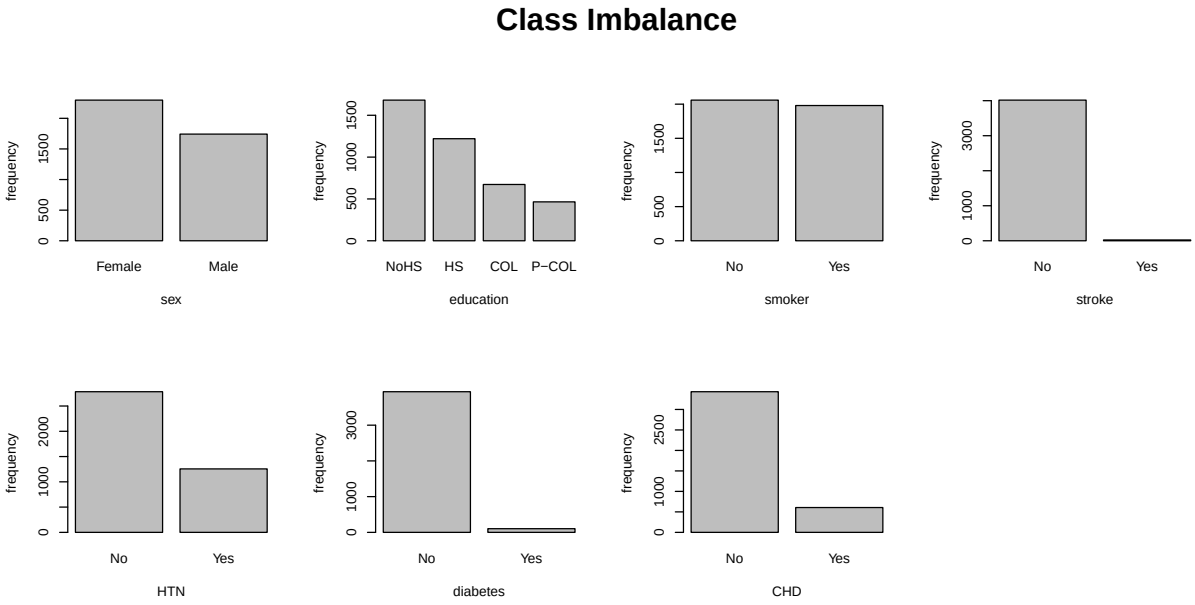
From the boxplots above we see that age might have a high discriminative power considering that the median age for those affected by CHD appears to be higher than that of of people not affected by it. The difference in medians of all other variables does not show any other significant difference between groups. However, it is noted that many outliers are present in most of the variables observed.

Now we can move on to plotting the main features of the categorical variables. However, it is also useful to quantify the association between categorical variables by using appropriate statistical tests as shown in the table below. The table summarizes the results of a Pearson's Chi-Squared test of independence carried out between each categorical variable in the dataset and CHD. It also adds the Cramer's Index of Association to check which variables are strongly associated with CHD. The predictors that show a small degree of association with CHD are hypertension (HTN), diabetes, and sex.

Table 2: Chi-Squared, Association Index, and p-value for Pearson's Test of Independence

	sex	education	smoker	stroke	HTN	diabetes
Chi-squared	30.56	31.25	1.62	8.74	126.28	31.39
Cramer's V	0.09	0.05	0.02	0.05	0.18	0.09
p-value	0.00	0.00	0.20	0.00	0.00	0.00

It is also useful to check how data are distributed in the different classes to understand if some categories are underrepresented or might lead later on to issues with imbalanced classes.



In the barplots above it is possible to determine the drastic class imbalances that are present in the dataset. Even the target variable (CHD) suffers from this issue. This highlights the fact that some conditions in the sample used for the study are particularly rare (i.e., stroke, diabetes).

4 Statistical Analysis

Now, before moving on to the model specification and fitting it is recommended to split the data into train and test sets to control for the issue of data leakage and to assess our model fairly on data it has never seen.

4.1 Train-Test Split

Due to the problem of imbalanced classes illustrated in **?@sec-categorical-variables**, the train-test split will be performed using `caret::createDataPartition()` to make sure the train and test samples are representative of the original one. The function `set.seed()` ensures that the result of the split are the same across different iterations.

```
set.seed(11)
train_idx <- caret::createDataPartition(df_cleaned$CHD, p=0.75)$Resample1

train_data <- df_cleaned[train_idx, ]
test_data <- df_cleaned[-train_idx, ]
```

4.2 Fitting Logistic Regression Model

Now that the data has been split, it is possible to fit the model on the training set.

```
lr <- glm(CHD ~ .,
          data = train_data,
          family = binomial)
```

Table 3: Summary Table for Logistic Regression Model

term	estimate	std.error	statistic	p.value
(Intercept)	-7.5466514	0.7710323	-9.7877236	0.0000000
sexMale	0.4385182	0.1193624	3.6738376	0.0002389
age	0.0704851	0.0072433	9.7311105	0.0000000
educationHS	-0.2376317	0.1365664	-1.7400456	0.0818510
educationCOL	-0.0814879	0.1613710	-0.5049725	0.6135781
educationP-COL	-0.0923956	0.1821199	-0.5073342	0.6119203
smokerYes	0.0655163	0.1741734	0.3761553	0.7068015
cpd	0.0243664	0.0067697	3.5993246	0.0003190
strokeYes	0.8940012	0.5662364	1.5788479	0.1143710
HTNYes	0.4623406	0.1398315	3.3064133	0.0009450
diabetesYes	1.0118432	0.2503772	4.0412751	0.0000532
chol	0.0020355	0.0012571	1.6191736	0.1054099
DBP	0.0125387	0.0056104	2.2348961	0.0254242
BMI	0.0054205	0.0138706	0.3907898	0.6959526
HR	-0.0016436	0.0045985	-0.3574121	0.7207833

The summary of the model shows that the main variables that have a significant effect on the log-odds of developing CHD are:

- Categorical variables (i.e., sex, HTN, diabetes): the coefficient represents the increase in the log-odds of developing CHD compared to the baseline group and keeping everything else constant.
- Numerical variables (i.e., age, cpd, DBP): the coefficient represents the increase in the log-odds of developing CHD when the predictor of interest increases by one unit and when everything else is kept fixed.

Variables whose coefficient is not statistically significant lack interpretability since we cannot say they are statistically different from 0 at the population level (i.e., no main effect has been found).

4.3 Fitting K-NN Model

In this section, different K-NN models are fitted using values of k ranging from 1 to 20. The performance of those models is discussed then in Section 4.4.

> **!** Important
>
> The numerical variables in the train and test subsets of the dataset have been rescaled using the `scale` function and have been used to train the different instances of K-NN models. The results have been assigned to the `train_scaled` and `test_scaled` variables, respectively.

```
X_train <- select(train_scaled, -CHD)
X_test <- select(test_scaled, -CHD)
y_train <- train_scaled$CHD
y_test <- test_scaled$CHD
```

4.4 Model performances

Now we can evaluate the accuracy of each model by producing a ROC curve for the Logistic Regression and by looking at the error metrics for different k parameters for the K-NN Classifier.

4.4.1 Confusion matrix

From the logistic regression model fitted above, we get the following predicted classes.

```
predics <- predict(lr, newdata = test_data, type = 'response')
classes <- ifelse(predics > .5, 'Yes', 'No')
```

Table 4: Confusion Matrix for Logistic Regression

	No	Yes
No	851	7
Yes	145	6

From the confusion matrix we can retrieve some metrics, including:

```
precision <- 6/(6+7)
recall <- 6/(6+145)
```

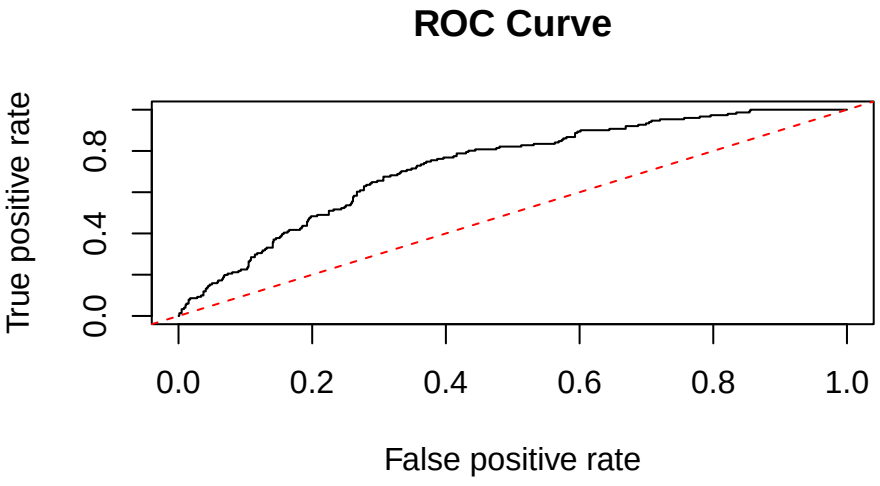
```
[1] "precision is: 0.46"
```

```
[1] "recall is: 0.04"
```

These metrics suggest that the model is not good enough for this classification task and that more advanced techniques should be used if one wants to improve on the current predictive capabilities of the model.

4.4.2 ROC Curve and AUC

The ROC curve can be used to show how well the Logistic Regression model fitted in Section 4.2 performs on unseen (test) data.



The overall AUC for this model is obtained with the following function from the `pROC` package.

```
# Compute ROC curve
roc_obj <- pROC::roc(test_data$CHD, lr_prob)

# Compute and print AUC
auc_value <- pROC::auc(roc_obj)
print(auc_value)
```

```
Area under the curve: 0.7274
```

Given the AUC value, one might be tempted to claim that the model has a decent discriminative power, but when this result is viewed in light of the accuracy metrics presented in Section 4.4.1 it becomes clear that the model does not perform well at all.

4.4.3 Evaluating different k for K-NN Classifier

Illustrated below are the error rates of the model for different values of k.

The best k parameter for the model, among those explored in the analysis and as evaluated on the test set, seems to be 20.

5 Discussion and Limitations

The current study might suffer from some limitations due to the class imbalances in the target value and in other categories as well.

5.1 K-NN vs Naive Classifier

We can also compare the K-NN model to a simpler (Naive) model that always predicts the absence of CHD.

```
naive_class <- rep('No', nrow(test_scaled))

acc_knn <- mean(naive_class != test_scaled$CHD)
```

```
[1] "The accuracy of the K-NN classifier is of 0.15"
```

We see that due to the imbalanced classes, even a model that learned nothing about the data is capable of producing an error rate of only 15%. This shows that the K-NN model is not good for this kind of problem. However, due to limitations in the page limit for this report, a strategy that was not explored was that of changing the distance metric used by the algorithm to find the top-k closest observations to the target. Employing a distance metric different from the Euclidean one could produce different and, possibly, better results.

5.2 Is the Logistic Regression Model better?

Put simply, the answer to this question is: not necessarily. The issue of imbalanced classes is a deep one and affects also the model fitted in Section 4.2. To improve on this preliminary result, it would be better to use a model that uses a different loss function. Specifically, a loss function that penalizes more the model for misclassifying the minority class compared to the majority class. By doing this, the model will have a greater chance of improving both its precision and recall as the number of true positives increases and the number of misclassification (both FP and FN) decreases.