Statistical Homework 1

Miglioranza Ettore

Firstly, load the data contained in *chd.csv*. With *summary()* obtain an overview of the data. We conclude that *sex* and *CHD* are stored as character variables, while the remaining predictors are in double-precision numeric format. Character variables may have to be converted to factors depending on the modelling approach. Next, we search for missing values.

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

Since we have a small data set to use, we think it is best to use an imputer to replace the missing values with the expected values. We make a copy of our data set, *data_simple*, on which we will apply the imputer function. We impute missing values with the basic metrics of the median for continuous data and mode for the ordinal categorical variable, *education*. Mode imputation is generally used when the distribution is highly imbalanced and the number of missing values is relatively small, making it acceptable to substitute with the most frequent category. In our case, the number of missing values is small, so it is a reasonable choice.

```r
simple_imputer <- function(data) {
  # Impute continuous variables with median
  data$cpd[is.na(data$cpd)]   <- median(data$cpd, na.rm = TRUE)
  data$chol[is.na(data$chol)] <- median(data$chol, na.rm = TRUE)
  data$BMI[is.na(data$BMI)]   <- median(data$BMI, na.rm = TRUE)
  data$HR[is.na(data$HR)]     <- median(data$HR, na.rm = TRUE)
  # Impute education with mode
  mode_edu <- as.numeric(names(which.max(table(data$education))))
  data$education[is.na(data$education)] <- mode_edu
  return(data)
```

```
}
data_simple <- simple_imputer(data)
```

Now that we have handled the missing values, we proceed to investigate potential significant relationships between variables. For this exploratory phase, we will analyze only the *data_simple* version of the dataset. To assess the discriminatory power of both continuous and categorical predictors, we examine their distributions across the different levels of the response variable (CHD). Specifically, we use boxplots for continuous variables and proportion-based bar plots for categorical ones. As shown in Figures 2 and 3, the distribution of *CHD* cases differ across various levels of predictors such as *Age*, *DBP*, *cpd*, *HTN*, *Sex* and *diabetes*. These visual differences provide preliminary evidence of a relationship between these predictors and the *CHD* outcome, supporting the hypothesis that they may contribute valuable information for *CHD* risk modeling.

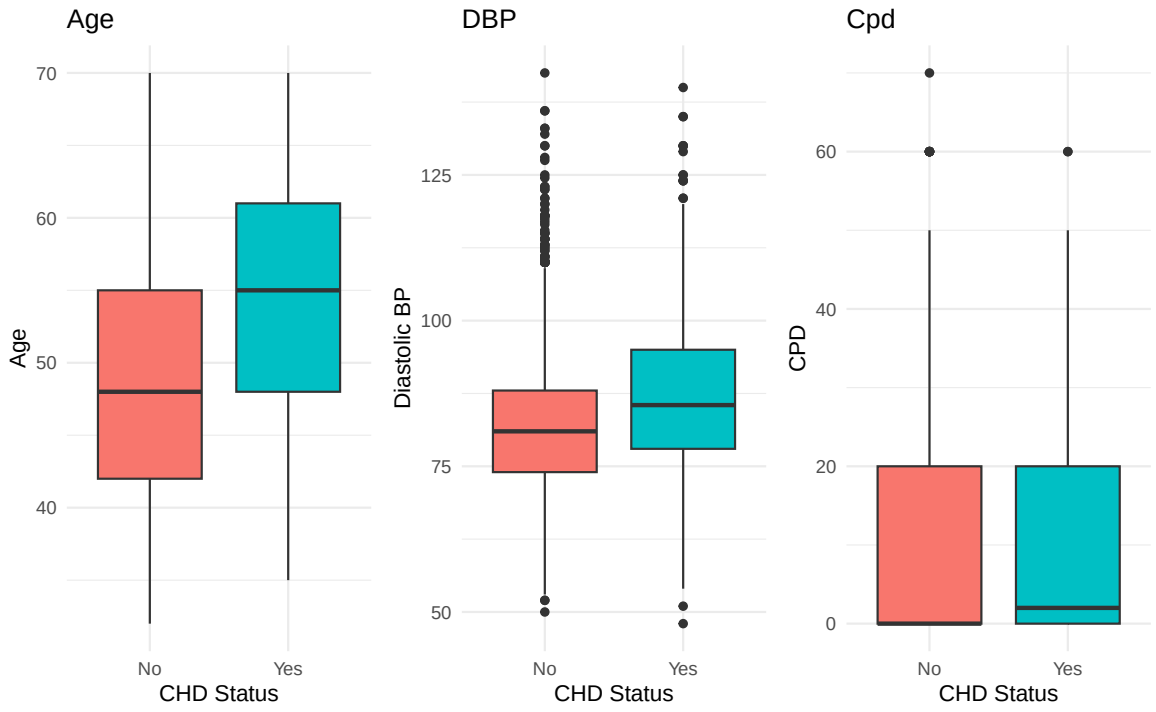

Figure 1: Distribution of continuous predictors by CHD status

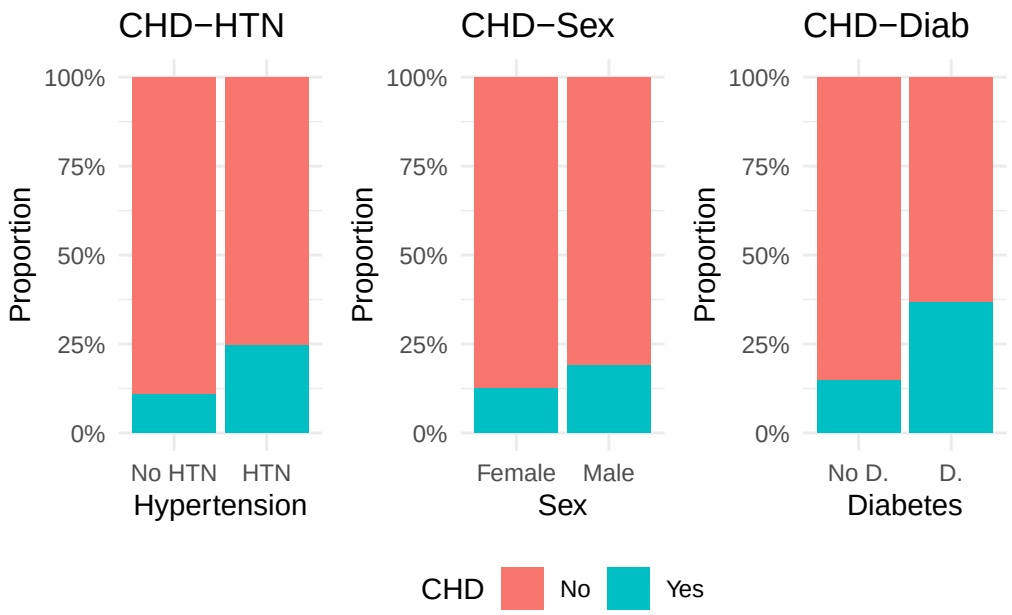

Figure 2: CHD proportions by hypertension, sex, and diabetes status

We proceed with the analysis of the nature of the response variable. Since the response variable is categorical and binary we want to assess the level of balance of the two categories by computing their proportions.

```
       No       Yes
0.8480415 0.1519585
```

The response variable is clearly imbalanced, with the majority of observations classified as "No" for CHD, meaning most individuals do not experience coronary heart disease within 10 years. This imbalance has important modeling implications—standard classifiers like logistic regression tend to favor the majority class, which can result in poor prediction performance on the minority class. In such cases, accuracy becomes a misleading evaluation metric, since predicting only the majority class can still yield a high overall accuracy (e.g., 84%) while missing all true positives. To ensure that the imbalance does not distort the training and test sets, we use **stratified sampling**, which maintains the original class proportions in both subsets. This is achieved in R using `caret::createDataPartition()` with a 70/30 split and a fixed seed (`42`). The resulting distributions closely match the full dataset: approximately 85% "No" and 15% "Yes" in both training and test sets.

```r
set.seed(42)
train_test_strata <- function(data) {
  # Create stratified split (e.g., 70% training)
  train_index <- createDataPartition(data$CHD, p = 0.7, list = FALSE)

  # Split the data
  train_data <- data[train_index,]
  test_data  <- data[-train_index,]

  return(list(train = train_data, test = test_data))
}

# Apply to both datasets and unpack
split_simple <- train_test_strata(data_simple)
train_simple <- split_simple$train
test_simple  <- split_simple$test
```

Logistic Regression Model

We fit the following GLM model:

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

Table 1: Logistic Regression Results

Predictor	Estimate	Std. Error	z value	Pr(>	z
(Intercept)	-7.4911	0.7688	-9.744	< 2e-16	***
sexMale	0.4967	0.1176	4.225	2.39e-05	***
age	0.0661	0.0071	9.341	< 2e-16	***
education	0.0143	0.0541	0.265	0.7908	
smoker	-0.0987	0.1694	-0.583	0.5599	
cpd	0.0216	0.0067	3.227	0.00125	**
stroke	0.9674	0.5347	1.809	0.0704	.
HTN	0.4457	0.1404	3.174	0.00150	**
diabetes	1.0892	0.2447	4.451	8.53e-06	***
chol	0.0019	0.0012	1.625	0.1042	
DBP	0.0148	0.0055	2.701	0.00692	**
BMI	-0.0071	0.0142	-0.505	0.6136	
HR	0.0024	0.0046	0.523	0.6013	

Interpretation of the Logistic Regression Model

The output summarizes a **logistic regression model** estimating the **log-odds** of developing **CHD** based on 12 predictors. The **Intercept (-7.491)** represents the log-odds for individuals in the reference categories (female, lowest education level, non-smoker, no stroke, no hypertension, no diabetes), indicating a **very low baseline probability** of CHD.

- **Sex (Male = 0.497)** is **highly significant** ($p < 0.001$); males have higher odds of CHD. Odds ratio: `exp(0.497) ~ 1.64`.
- **Age (0.066)** is **significant** ($p < 0.001$); each additional year increases CHD odds by ~6.8%.
- **Education (0.0143)** is **not significant** ($p = 0.791$); it has little effect on CHD risk.
- **Smoker status (-0.099)** is **not significant**, but **cpd (0.0216)** is **significant** ($p = 0.001$), indicating smoking intensity is more predictive. Odds ratio: `exp(0.0216) ~ 1.022`.
- **Stroke (0.967)** is **marginally significant** ($p = 0.070$), with individuals having stroke history over **2.6 times the odds** of CHD.
- **HTN (0.446)** is **significant** ($p = 0.0015$), raising CHD odds by ~56%.
- **Diabetes (1.089)** is **highly significant** ($p < 0.001$); diabetics have nearly **three times the odds** of CHD.
- **Cholesterol (0.00194)** is **not significant** ($p = 0.104$), suggesting a weak effect.
- **DBP (0.0148)** is **significant** ($p = 0.0069$); each mmHg increase raises CHD odds by ~1.5%.
- **BMI (-0.0071)** and **HR (0.0024)** are **not significant**, indicating minimal predictive power.

Overall, variables like sex, age, cpd, stroke, HTN, diabetes, and DBP show meaningful associations with CHD, while others contribute little when adjusting for these effects.

K-NN Classifier

Since we know that models based on clustering perform poorly with features on different scales, we standardize all continuous variables before the fitting process. Without standardization, features like *cholesterol* or *age* could dominate the distance metric simply due to their larger numeric ranges. We extract the mean and standard deviation from the training set and use them to standardize both the training and test sets. After this, we follow the procedure below to fit a K-NN model: set up a training control with 5-fold, cross-validation, apply grid search to fine-tune the k parameter (using a tuning grid from $k = 5$ to $k = 30$) and fit the model on the standardized data. We evaluate model performance using the accuracy metric, selecting the value of k that yields the highest cross-validated accuracy.

```r
# Set up training control with 5-fold CV
ctrl <- trainControl(method = "cv", number = 5)
# Define tuning grid for k (number of neighbors)
k_grid <- expand.grid(k = 5:30)
# Fit the k-NN model
set.seed(42)
knn_model <- train(CHD ~ ., data = train_scaled, method = "knn",
                   tuneGrid = k_grid, trControl = ctrl)
```

```
Highest accuracy of 0.8483  with k = 25 .
```

Performance evaluation

The next step is to evaluate the models. Given the nature of the response variable, it is clear that *accuracy* alone is not an appropriate evaluation metric. In particular, since this is a medical study, we are especially concerned with not missing high-risk patients, while we are more tolerant of issuing a false alarm. In this context, a more meaningful evaluation metric is the *FNR* (False Negative Rate), which measures the proportion of patients who developed *CHD* but were not identified by the system. The *FNR* is defined as:

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - \text{Sensitivity}$$

Logistic Regression metrics:

```
Accuracy: 0.8513 Sensitivity: 0.0415 Specificity: 0.9963 FNR: 0.9585
```

K-NN metrics:

```
Accuracy: 0.8482 Sensitivity: 0 Specificity: 1 FNR: 1
```

The logistic regression is strongly skewed towards the prediction of the majority class ('No CHD') due to the imbalance. Although it performs slightly better than chance (as reflected in the AUC), it struggles with the minority class. On the other hand, k-NN can preserve accuracy through correct classification of the dominant class, but it is entirely incapable of detecting minority-class cases without further balancing strategies.

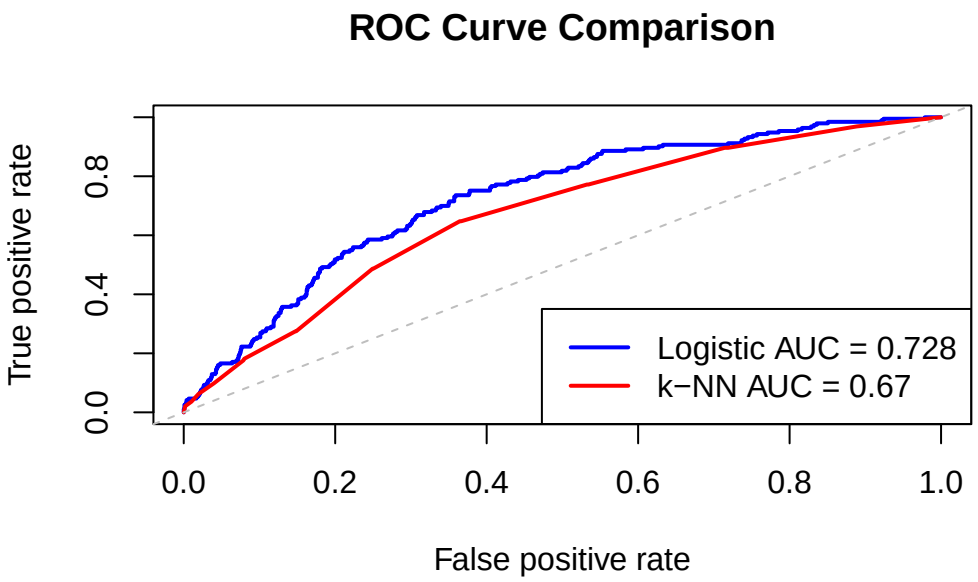

Figure 3: ROC AUC Logistic regression vs. K-NN

Conclusion

The two models show comparable performance in terms of overall *accuracy*, but focusing on the *False Negative Rate (FNR)*—a key metric in medical applications—*logistic regression* outperforms *k-NN*, which fails to detect any true positives. Despite still exhibiting a high FNR, the logistic model offers a modest improvement in identifying high-risk patients. However, using classification models without explicitly addressing class imbalance is suboptimal. To improve performance in such contexts, it is essential to incorporate techniques such as resampling, penalisation for misclassifying the minority class, threshold adjustments that favor sensitivity, or asymmetric cost functions. The analysis is also subject to several limitations. Firstly, missing values were handled using simple imputation, which could introduce bias if the missingness mechanism is not random. Secondly, as previously discussed, the outcome variable is highly imbalanced, with approximately 85% of observations corresponding to the absence of *CHD*, leading to a tendency for models to favor the majority class. Additionally, potential multicollinearity among predictors may distort coefficient estimates and reduce model interpretability. Lastly, the dataset lacks relevant features such as dietary habits, alcohol consumption, physical activity, or genetic predisposition, which may limit the overall predictive power of the models.

Bonus

To solve the imbalance problem, we want to use the SMOTE technique to see if the two models improve. SMOTE (*Synthetic Minority Over-sampling Technique*)[1] is a very powerful technique that generates synthetic examples instead of duplicating existing ones. SMOTE generates new minority class samples by interpolating between an existing minority point and one of its nearest minority neighbors. It picks a neighbor at random and creates a new point somewhere along the line between them in feature space. In our case, since we have mixed data types, we have to use SMOTE NC (Nominal-Continous). We set the seed to *42* again and re-train the models. Apparently, SMOTE worked well, greatly improving the *sensitivity* of both models. I can't show the code for space issue, but it can be found on my git[2].

```
 Logistic Regression with SMOTE
```

```
          Reference
Prediction  No Yes
       No  848  93
       Yes 230 100
```

```
Accuracy: 0.7459 Sensitivity: 0.5181 Specificity: 0.7866 FNR: 0.4819
```

```
 K-NN with SMOTE (best k = 5 )
```

```
          Reference
Prediction  No Yes
       No  762 113
       Yes 316  80
```

```
Accuracy: 0.6625 Sensitivity: 0.4145 Specificity: 0.7069 FNR: 0.5855
```

[1] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. https://doi.org/10.1613/jair.953

[2] git rep with .qmd: https://github.com/ettoremiglioranza1012/StatMods_HWdir.git