Statitical Models Homework 1

Exploratory Data Analysis

After setting up a virtual environment suitable for carrying out the scientific analysis, we load some useful packages to improve the capabilities of the analysis software. In particular, as suggested we load *ISLR*, *ISLR2*, *ROCR* and *tidyverse*. For the purposes of this research, we will also use *e1071*, *caret*, *corrplot*, *patchwork*, *class*.

The first step in our analysis is to load the data contained in *chd.csv*. With the function *glimpse()* obtain an overview of the data.

```
Rows: 4,238
Columns: 13
$ sex       <chr> "Male", "Female", "Male", "Female", "Female", "Female", "Fem~
$ age       <dbl> 39, 46, 48, 61, 46, 43, 63, 45, 52, 43, 50, 43, 46, 41, 39, ~
$ education <dbl> 4, 2, 1, 3, 3, 2, 1, 2, 1, 1, 1, 2, 1, 3, 2, 2, 3, 2, 2, 2, ~
$ smoker    <dbl> 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, ~
$ cpd       <dbl> 0, 0, 20, 30, 23, 0, 0, 20, 0, 30, 0, 0, 15, 0, 9, 20, 10, 2~
$ stroke    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ HTN       <dbl> 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, ~
$ diabetes  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ chol      <dbl> 195, 250, 245, 225, 285, 228, 205, 313, 260, 225, 254, 247, ~
$ DBP       <dbl> 70.0, 81.0, 80.0, 95.0, 84.0, 110.0, 71.0, 71.0, 89.0, 107.0~
$ BMI       <dbl> 26.97, 28.73, 25.34, 28.58, 23.10, 30.30, 33.11, 21.68, 26.3~
$ HR        <dbl> 80, 95, 75, 65, 85, 77, 60, 79, 76, 93, 75, 72, 98, 65, 85, ~
$ CHD       <chr> "No", "No", "No", "Yes", "No", "No", "Yes", "No", "No", "No"~
```

From the output, we observe that the variables *sex* and *CHD* are stored as character variables, while the remaining predictors are in double-precision numeric format. This suggests that most of the variables are already in a format suitable for analysis, although the character variables may have to be converted to factors depending on the modelling approach.

First of all, let us proceed with some basic checks on the data. The wise thing to do first is always to look for missing values.

sex	age	education	smoker	cpd	stroke	HTN	diabetes
0	0	105	0	29	0	0	0
chol	DBP	BMI	HR	CHD			
50	0	19	1	0			

Since we have a small data set to use, we think it is best to use an imputator to replace the missing values with the actual expected values. We will now make a copy of our data set, *data_simple*, on which we will apply the imputer function. We impute missing values with basic metrics such as mean, median (if the data distribution is skewed) for continuous data and mode for the categorical ordinal *education*. Mode imputation is generally reasonable 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. However, mode imputation ignores the ordinal nature of the variable and does not take into account potential correlations with other variables. In our case, the number of missing values is small, so these limitations are unlikely to significantly affect the results. The output below confirms that the missing values were correctly replaced by the imputer function.

sex	age	education	smoker	cpd	stroke	HTN	diabetes
0	0	0	0	0	0	0	0
chol	DBP	BMI	HR	CHD			
0	0	0	0	0			

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 differs 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.

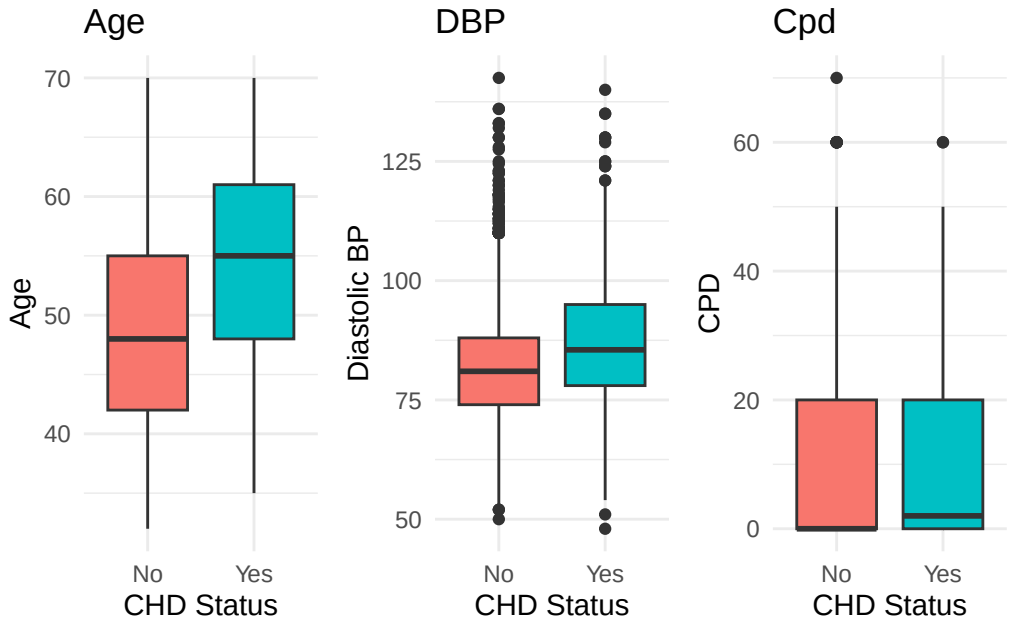

Figure 1: Distribution of continuous predictors against CHD

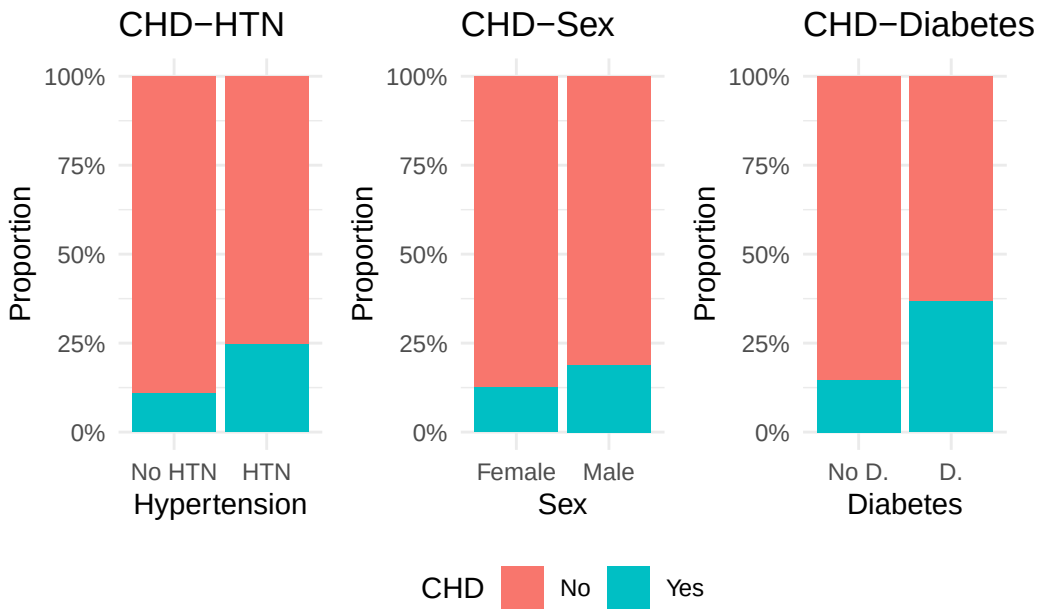

Figure 2: CHD proportions by hypertension, sex, and diabetes status

We proceed with the analysis of the nature of the response variable. Since the response variable is categorical and binary we want to asses the level of balance inside the two categories by computing the proportion of each one of them on the total number of observations.

```
       No       Yes
0.8480415 0.1519585
```

The output clearly shows that the response variable is strongly skewed toward the "No" class. This means that most observations do not have the event of interest (e.g., no coronary artery disease within 10 years). This has some implications for modeling, such as the fact that standard classifiers, e.g., logistic regression, will likely favor the most representative class, and perform poorly in making predictions about the minority class. In this scenario, the Accurancy is misleading, since the prediction of the majority class alone would still provide high accuracy (e.g., 84%) even if all cases of the minority class were missing.

In addition, we must prevent the random partitioning of data into a training and test set from leading to unbalanced sets where the true proportions of the sample are not actually represented correctly. To do this we use a technique called stratified sampling, which is aimed precisely at maintaining the proportions of the categories of a variable in both subsets of the sample. In practice, we will divide the sample into two groups filtered by the two categories of the response variable (in this case "yes" and "no"). These two groups are called *strata*. We then branch out some instances from the two groups and place them in our desired training or test set in a certain proportion (we chose to put 70% of the instances in the training set and the other 30% in the test). This maintains the original proportion in both sets. In R, we use *caret::createDataPartition()* which is the function provided to do this, after setting the seed to *42*. After that, we verify the new proportions of both the 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)
```

```r
train_simple <- split_simple$train
test_simple  <- split_simple$test
```

```
       No       Yes
0.8479946 0.1520054
```

```
       No       Yes
0.8481511 0.1518489
```

Logistic Regression Model

We fit the following GLM model:

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

```
Call:
glm(formula = CHD ~ sex + age + education + smoker + cpd + stroke +
    HTN + diabetes + chol + DBP + BMI + HR, family = binomial,
    data = data_glm)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.491146   0.768762  -9.744  < 2e-16 ***
sexMale      0.496706   0.117565   4.225 2.39e-05 ***
age          0.066061   0.007072   9.341  < 2e-16 ***
education    0.014336   0.054053   0.265  0.79084
smoker      -0.098736   0.169360  -0.583  0.55990
cpd          0.021571   0.006684   3.227  0.00125 **
stroke       0.967427   0.534693   1.809  0.07040 .
HTN          0.445656   0.140401   3.174  0.00150 **
diabetes     1.089238   0.244692   4.451 8.53e-06 ***
chol         0.001943   0.001196   1.625  0.10421
DBP          0.014842   0.005495   2.701  0.00692 **
BMI         -0.007149   0.014157  -0.505  0.61360
HR           0.002397   0.004588   0.523  0.60129
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2528.9  on 2966  degrees of freedom
Residual deviance: 2267.2  on 2954  degrees of freedom
AIC: 2293.2

Number of Fisher Scoring iterations: 5
```

Interpretation of the Logistic Regression Model

The output summarizes the results of a fitted **logistic regression model** estimating the **log-odds** of developing **CHD** (Coronary Heart Disease) as a function of an intercept and 12 predictor variables.

- The estimated **Intercept (-7.491)** represents the log-odds of developing CHD for individuals in all the **reference categories** (e.g., female, lowest education level, non-smoker, no stroke, no hypertension, no diabetes). Its strongly negative value reflects a very **low baseline probability** of developing CHD in this subgroup.
- The coefficient for **sexMale (0.497)** is **positive and highly significant** ($p < 0.001$), suggesting that, all else equal, **males have higher odds** of developing CHD compared to females. The odds ratio is `exp(0.497)  1.64`.
- **Age (0.066)** is also **highly significant** ($p < 0.001$), with each additional year of age increasing the log-odds of CHD. The odds ratio is `exp(0.066)  1.068`, indicating a **6.8% increase in odds per year**.
- The variable **education (0.0143)** is **not statistically significant** ($p = 0.791$), suggesting that, in the presence of other covariates, education level does not have a meaningful impact on CHD risk.
- **Smoker status (-0.099)** is **not significant** ($p = 0.560$), indicating that being a smoker vs. non-smoker may not be predictive of CHD **when controlling for smoking intensity (cpd)**.
- In contrast, **cpd (cigarettes per day; 0.0216)** is **statistically significant** ($p = 0.001$), suggesting that the **intensity of smoking** is more predictive of CHD risk than smoker status alone. The odds ratio is `exp(0.0216)  1.022`, meaning a **2.2% increase in odds per cigarette per day**.
- **Stroke (0.967)** has a **marginally significant effect** ($p = 0.070$), with an odds ratio of `exp(0.967)  2.63`, suggesting that individuals with a history of stroke have **more than twice the odds** of developing CHD.

- **HTN (0.446)** is **statistically significant** ($p = 0.0015$), with an odds ratio of `exp(0.446)` $\approx$ `1.56`, indicating that individuals with hypertension have **56% higher odds** of CHD.
- **Diabetes (1.089)** is **highly significant** ($p < 0.001$), with an odds ratio of `exp(1.089)` $\approx$ `2.97`, meaning diabetics have nearly **three times the odds** of developing CHD compared to non-diabetics.
- **Cholesterol (0.00194)** is **not statistically significant** ($p = 0.104$), indicating a **weak or inconclusive** effect on CHD risk in this model.
- **DBP (Diastolic Blood Pressure; 0.0148)** is **statistically significant** ($p = 0.0069$), with an odds ratio of `exp(0.0148)` $\approx$ `1.015`, meaning each 1 mmHg increase in DBP is associated with a **1.5% increase** in the odds of CHD.
- **BMI (-0.0071)** and **HR (Heart Rate; 0.0024)** are **not statistically significant** ($p = 0.614$ and $p = 0.601$, respectively), suggesting they do not contribute meaningfully to the prediction of CHD in this model.

K-NN Classifier

Since we know that models based on clustering perform poorly with features with different scale, before the fitting process we standardize all the continuos variable. Without standardization, features like cholesterol or age might dominate the distance metric simply due to larger numeric ranges. We extract the mean and s.d. from the train set, and we use them to standardize both the train and test set. After this, we follow this procedure to fit a *K-NN* model: we set up a training control with a 5-fold Cross Validation, we apply grid search to fine tune the k parameter (the tuning grid for k will be *[5,30]*) and finally we fit the model. We produce an output showing the best value of k and the accurancy result.

```
# Set up training control with 5-fold CV
set.seed(42)
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
)
# Show best value of k and accuracy results
# Extract best k and corresponding accuracy
best_k <- knn_model$bestTune$k
best_acc <- knn_model$results %>%
  filter(k == best_k) %>%
  pull(Accuracy)

# Print summary
cat("Highest accuracy of", round(best_acc, 4), " with k =", best_k, ".\n")
```

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

Performance evalutation

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:

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

We produce the confusion matricies of both models to extract this metric:

```
     Actual
LR      No  Yes
  No  1074  185
  Yes    4    8
```

```
Logistic Regression Accuracy: 0.8513
```

```
Logistic Regression FNR: 0.9585
```

```
     Actual
KNN     No  Yes
  No  1078  193
  Yes    0    0
```

```
k-NN Accuracy: 0.8482
```

```
k-NN FNR: 1
```

Conclusion

In conclusion, the two models exhibit comparable performance in terms of overall *accuracy*. However, considering the metric of primary interest—the *False Negative Rate (FNR)*—the *logistic regression* model outperforms *k-NN*. Specifically, the *k-NN* model fails to correctly identify any true positive cases, misclassifying all actual *CHD* cases as negative. Although the *logistic regression* model also demonstrates a high false negative rate, incorrectly classifying approximately 96% of the true positives, it nonetheless provides a modest improvement over *k-NN* in identifying high-risk patients. Consequently, in an application where correctly identifying the minority class is crucial, *logistic regression* appears to be the more suitable model. The first limitation we wish to address concerns some missing values that were present in the dataset and were handled by simple imputation techniques. However, if the mechanism of missingness is not well captured by the distribution of the observed data, this may introduce bias into the analysis. Secondly, a major limitation is the imbalance in the outcome variable: approximately 85% of the observations correspond to the absence of *CHD*. This imbalance can lead the models to favor the majority class, thereby increasing the risk of overlooking high-risk individuals.Another limitation relates to potential multicollinearity among predictors, as correlations between key variables may distort coefficient estimates and reduce model interpretability. Additionally, some predictors, such as *diabetes*, occur infrequently in the dataset. This low prevalence can result in unstable coefficient estimates and limit the variable's discriminative power. Finally, the dataset may omit relevant risk factors such as dietary habits, alcohol consumption, physical activity, or genetic predisposition. The absence of these variables could reduce the predictive accuracy of the models and overlook important contributors to *CHD* risk.