Homework1: classification models on CHD data set

Camilla Bonomo matr. 255138

2025-04-01

This paper analyses data on Coronary Heart Disease and aims to evaluate two classification models for predicting CHD status from a set of demographic and clinical predictors. The compared models are Logistic Regression (LR) and K-Nearest Neighbors (k-NN). The main objective of this paper is to understand the influence of predictors on CHD and compare performance metrics (accuracy, sensitivity, specificity, AUC) across Linear regression and k-NN. (link to repository: https://github.com/camillabonomo02/Homework_1.git)

Data exploration and cleaning

For this analysis the data set that has been used - defined here as *df_chd* - contains individual-level observations on CHD status, demographics (e.g., age, sex, education), and clinical parameters (e.g., heart rate, cholesterol).

In particular, the *CHD* variable is the **categorical response** variable that returns a binary outcome ("No" or "Yes"). The predictors are the other 12 variables included in the data set, which are a mix of continuous and categorical variables. Having said that, it is important to factorize all the categorical variables so that they can be correctly recognized and processed by the classification algorithms, enabling accurate parameter estimation and interpretation.

Then, a quick research for NAs is done returning the following:

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			

For the continuous variables it has been decided to impute them with the *mean*. The exception is made for the variable **cpd** whose NAs are imputed with the *median* as its distribution may

be skewed toward zero. Meanwhile, the NAs values of the categorical variable education are imputed with the *mode*.

Another main point for the data exploration is to asses how well each predictor in the data set separates the two CHD classes (CHD = Yes/No) to determine whether a given variable appears to have discriminative power for CHD status.

For continuous predictors *boxplots* are used which provide a quick visual understanding of whether the distributions differ across CHD classes.

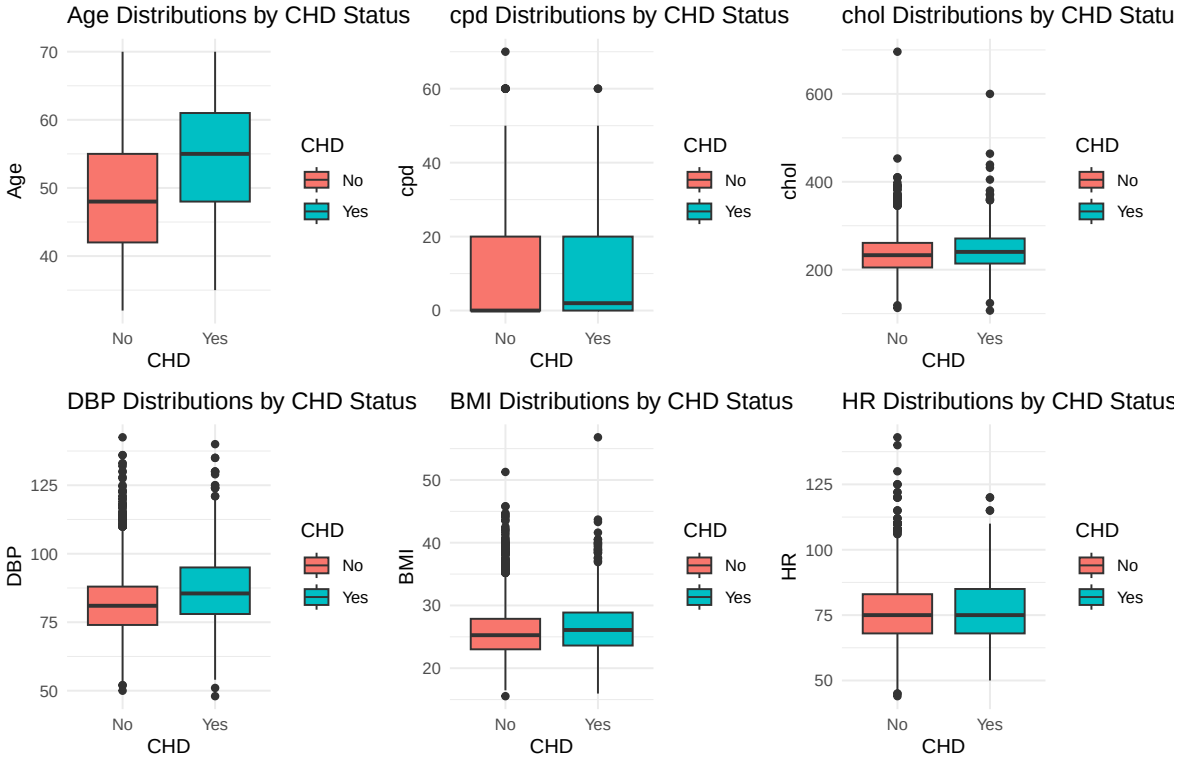

Figure 1: Plots for assessing numerical variables discriminative power for CHD

Those boxplots highlight, by their distributions, how variables like *age* can have a high discriminative power over CHD.

For categorical predictors, it is insightful to compare the CHD proportions across categories through *bar plots* showing how frequently CHD appears in each level of the factor.

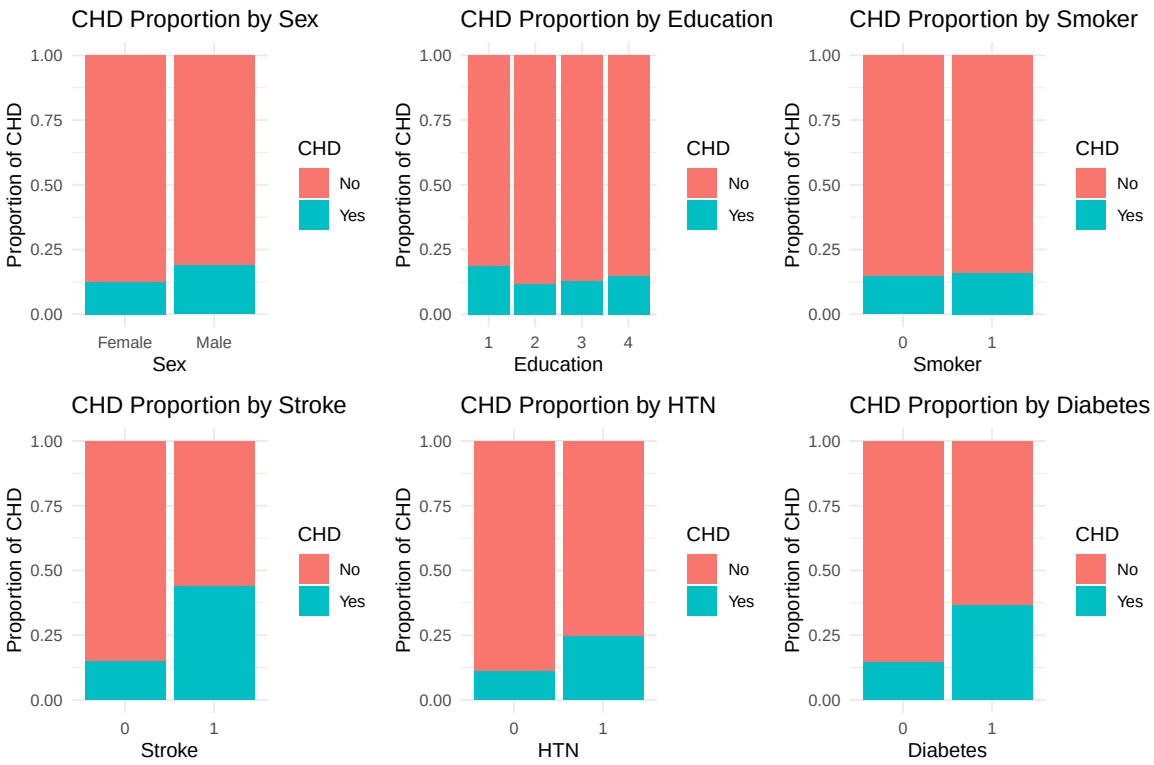

Figure 2: Plots for assessing categorical variables discriminative power for CHD

It is possible to notice how variables like *diabetes*, *stroke* and *HTN* can have a high discriminative power over CHD.

Splitting data

A stratified split was used to partition the data set into training (70%) and test (30%) subsets while preserving the proportion of CHD and non-CHD cases, that have the following imbalanced distribution:

CHD	Freq
No	3594
Yes	644

The partition then returns a train (70%) and test set (20%) with the class distributed as 85% for "No" and 15% for "Yes".

```r
set.seed(123)
index <- caret::createDataPartition(df_chd$CHD, p = 0.7,
    list = FALSE)
train_data <- df_chd[index, ]
test_data <- df_chd[-index, ]
```

Fitting the Logistic Regression model

We now want to predict the *CHD* target variable using a multivariate Logistic Regression model. We fit a logistic regression of the form:

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

as follows:

```r
glm_fit <- glm(CHD ~ ., data = train_data, family = binomial(link = "logit"))
```

Here is the summary of the model fit:

term	estimate	std.error	statistic	p.value
(Intercept)	-7.0602105	0.7528540	-9.3779281	0.0000000
sexMale	0.3713356	0.1191821	3.1157003	0.0018351
age	0.0643635	0.0071881	8.9541714	0.0000000
education2	-0.2120292	0.1370685	-1.5468843	0.1218911
education3	-0.1564499	0.1654120	-0.9458196	0.3442406
education4	-0.0848605	0.1857248	-0.4569155	0.6477318
smoker1	-0.0388205	0.1698394	-0.2285717	0.8192019
cpd	0.0259359	0.0066813	3.8818788	0.0001037
stroke1	1.6898842	0.5741733	2.9431605	0.0032488
HTN1	0.5443209	0.1384353	3.9319527	0.0000843
diabetes1	0.7523745	0.2503335	3.0054890	0.0026515
chol	0.0011392	0.0012026	0.9472817	0.3434953
DBP	0.0124533	0.0054584	2.2814905	0.0225194
BMI	0.0028929	0.0139641	0.2071696	0.8358774
HR	0.0006075	0.0045684	0.1329781	0.8942107

By looking at the table it possible to notice that several predictors have p-values well below 0.05, indicating that they significantly affect the odds of CHD in the presence of the other variables, especially age, cpd and HTN1. These values suggest, respectively, that: as *age* increases, the log-odds of CHD rise, suggesting *strong age dependence*; the more *cigarettes smoked per day* (cpd), the higher the log-odds of CHD; Hypertension (HTN1) is a strong and statistically significant contributor to CHD risk.

Some variables (e.g., education levels, smoker1, chol, BMI, HR) do not reach statistical significance in this model. This lack of significance does not necessarily mean they have no effect, but rather that any effect is not detected given the current data and model specification.

Evaluation of the model

The values of the variable CHD are predicted by applying the fitted Logistic model and setting the threshold to *0.3*. This value for threshold has been decided by taking into account the imbalance between "Yes" (15%) and "No"(85%) values in the target variable CHD. In an imbalanced scenario a threshold of 0.5 tends to predict mostly "No" and misses many of the minority-class positives. By reducing the threshold, the model is allowed to flag more "Yes" cases:

```
glm_prob <- predict(glm_fit, newdata = test_data, type = "response")
glm_pred <- ifelse(glm_prob > 0.3, "Yes", "No")
```

It is now possible to evaluate the model. For generating the confusion matrix and estimating the necessary metrics we use **confusionMatrix()** that calculates a cross-tabulation of observed and predicted classes with associated statistics.

```
cm_glm <- confusionMatrix(factor(glm_pred, levels = c("No",
    "Yes")), test_data$CHD)
```

The following confusion matrix is returned:

```
          Reference
Prediction  No Yes
       No  995 153
       Yes  83  40
```

This table is the perfect tool for computing some metrics extracted as follows:

```
accuracy_glm <- cm_glm$overall["Accuracy"]
sensitivity_glm <- cm_glm$byClass["Sensitivity"]
specificity_glm <- cm_glm$byClass["Specificity"]
```

Accuracy	Sensitivity	Specificity
0.814	0.923	0.207

It has been noticed that lowering the logistic threshold to 0.3 yields an *accuracy* of *0.814*. Sensitivity or True Positive Rate is high (0.9230); specificity is just 0.2073, implying the model still misses many "Yes" cases. In this case FPR (1-specificity) is high (0.7927).

ROC curve

From the computations that has been made it is possible to obtain the ROC curve for evaluating the overall model. For doing that it is more practical to use the **roc** function as follows:

```
glm_roc <- roc(response = test_data$CHD, predictor = glm_prob,
    levels = c("No", "Yes"), direction = "<")
```

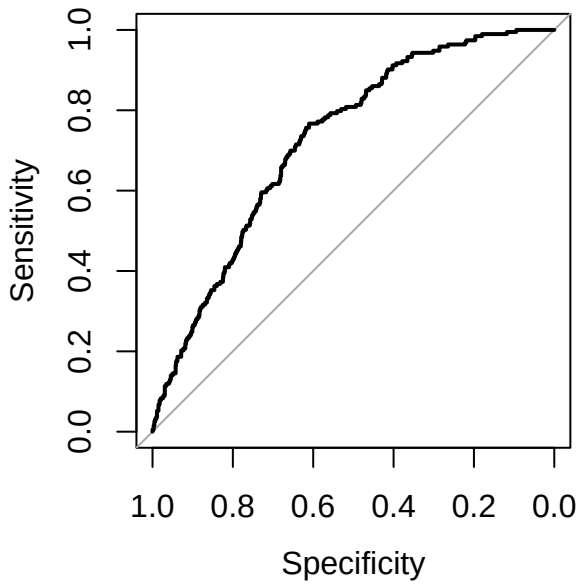

Figure 3: ROC Curve - Logistic Regression (Threshold=0.3)

From a graphical standpoint it is possible to notice that the performance of our model is moderate. Moreover, to assess this it is also possible to compute the AUC which is:

Area Under the Curve
0.7253069

This value states that the model is correctly ranking positive vs. negative cases about 70% of the time. This suggests that the model has some real predictive value but could likely be improved.

Fitting the K-Nearest Neighbors model

For KNN, the scale of a variable is very important, as KNN is based on distances to identify observations near to each other. So, variables on a large scale will impact the distance way more than variables on a smaller scale. Only the numeric variables are *scaled* and then the KNN model is trained with the new scaled data.

```
numeric_vars <- c("age", "cpd", "chol", "DBP", "BMI",
    "HR")
pre_proc <- preProcess(train_data[, numeric_vars],
    method = c("center", "scale"))

train_scaled <- train_data
test_scaled <- test_data

train_scaled[, numeric_vars] <- predict(pre_proc, train_data[,
    numeric_vars])
test_scaled[, numeric_vars] <- predict(pre_proc, test_data[,
    numeric_vars])
```

For tuning the k parameter, given a data set of 4238 observation, it has been chosen to try with a wide range (from 1 to 100) to make sure to avoid underfit or overfit of the model. Moreover, stepping by 2 (e.g., seq(1, 100, by = 2)) is common for binary classification so that ties are less likely. As regards the cross-validation a 10-fold repeated CV is optimal for balancing bias and variance (repeating 3 times further stabilizes performance estimates).

```
ctrl <- trainControl(method = "repeatedcv", number = 10,
    repeats = 3)

tune_grid <- expand.grid(k = seq(1, 100, by = 2))

knn_fit <- train(CHD ~ ., data = train_scaled, method = "knn",
    trControl = ctrl, tuneGrid = tune_grid)
```

Many trials with different ranges has been made and in most cases the optimal k has resulted as the following:

optimal k
19

This value of k is the one that minimizes the value of the error rate. In fact, *caret* internally performs cross-validation over various k values and picks the best based on a chosen metric.

Evaluation of the model

For evaluating the performance of this model and getting the confusion matrix it is exploited the same method used for the Logistic Regression.

```
knn_pred <- predict(knn_fit, newdata = test_scaled)

cm_knn <- confusionMatrix(knn_pred, test_scaled$CHD)
```

The following confusion matrix is returned:

```
          Reference
Prediction   No  Yes
       No  1077  193
       Yes    1    0
```

By looking at the table it is already clear that the model KNN does not detect any of the minority "Yes" instances under these settings. Moreover, the metrics confirm this:

Accuracy	Sensitivity	Specificity
0.847	0.999	0

The K-NN model overall *accuracy* is *0.8474*. Sensitivity (0.9991) is extremely high; specificity (i.e., correctly identifying "Yes") is *0.0000*, meaning the model never correctly classifies a "Yes" case.

Conclusions and final considerations

Based on the results, Logistic Regression (LR) appears to be more suitable than K-Nearest Neighbors (KNN) for predicting Coronary Heart Disease (CHD) in this context. While KNN achieved slightly higher overall accuracy, it completely failed to identify any positive CHD cases (specificity = 0), making it ineffective for detecting the minority class. LR, on the other hand, demonstrated better balance, with a reasonable trade-off between sensitivity (0.923) and specificity (0.207), and an AUC of ~0.73 indicating moderate predictive power. Key predictors identified included age, hypertension, and cigarettes per day. However, the study's limitations include class imbalance, potential overfitting in KNN, and lack of external validation. More robust models or re-sampling techniques could improve performance in future work.