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*, *mice*, *caret*,

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 represented as character variables, while the remaining predictors are stored as double-precision numerical values (real numbers). 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 the first copy of our data set, as we will apply two different imputation procedures to the two copies. We will then compare the future results of the analysis on these two different batches of data to see if one set of techniques worked better than the other.

We then create *data_simple*, where 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*. We then copy the dataset into *data_mice*, a copy in which we will use more complex functions.

Starting with the *simple_imputer()* function, we first want to evaluate the skewness of continuous numeric variables.

```
      cpd      chol       BMI        HR
1.2470206 0.8707979 0.9812762 0.6440255
```

Using the rule of thumb, we can asses moderate skewness in all variables, except for *cpd*, which proves to be significantly skewed. Therefore, the median will be a preferable metric for imputing values.

For simplified imputations, we will use mode to complete missing values for *education*. Mode imputation is commonly used for the categorical variables but has some limitations. It 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. However, we will evaluate their impact by comparing the performance of this basic dataset with that imputed by a more advanced method.

```
      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
```

For complex imputation we use *mice* (Multivariate Imputation by Chained Equations). Mice has the ability to perform multiple imputations by actually predicting each variable with missing data, using the other variables as predictors (which is great because this mitigates the uncertainty of missing values). FORSE APPROFONDISCI FUNZIONI USATE!

```
      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
```

As before, R's output shows that the imputations occurred correctly. Now that we have handled the missing values well, we will analyze the nature of the response variable.

```
       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. Moreover, 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 (e.g., 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.

```
       No       Yes
0.8479946 0.1520054

       No       Yes
0.8481511 0.1518489


       No       Yes
0.8479946 0.1520054

       No       Yes
0.8481511 0.1518489
```