Labwork on *Statistical learning*

Probabilities and statistics - SNS Master - University of Montpellier

Nicolas Sutton-Charani – Euromov DHM – IMT Mines Alès

Solutions will be given in this online notebook.

green = bonus questions
blue = package names
purple = function names

1. **Data import**
    a. Import the *framingham* dataset from *here*.
    b. Compute the dimensions of the *framingham* dataset and the type of each variables.
    c. Draw the distribution of the BMI variable through a boxplots and a histograms.
    d. Draw the distribution of the TenYearCHD variable through barplots.

2. **Missing data imputation**
    a. Compute the number of missing data per variable.
    b. Impute the missing data by mean for numerical variables.
    c. Impute the missing data by the most frequent values for binary variables.

3. **Normalisation :** normalise each numerical variables so that it lies in [0, 1].

For the following, the ***x*** matrix will refer to the normalised completed numerical variables of the *framingham* dataset.

4. **Correlations**
    a. Compute the Spearman correlation matrix on ***x***.
    b. Plot the corresponding heatmap correlation
    c. Redo the previous question including binary variables of the *framingham* dataset with the following customisation : plot only the a triangular correlation heatmap

with 45° rotated x-axis variable names, set negative correlations to blue and positive ones to red.

5. PCA

a. With the *PCA* function of the *FactoMineR* package compute a PCA on normalised numerical variables.
b. With the *fviz_functions*, plot the the percentage of explained variance per dimension and the 2 PCA main plots (1 for examples and 1 for variables) and comment.
c. Plot the contributions of each variable to the 5 first dimensions with the *get_pca_var* and *corrplot* functions of the *corrplot* package.
d. Recompute the individual PCA plot and successively color points according to the binary variables " TenYearCHD" and " prevalentHyp ".

6. Clustering

a. With the *fviz_nbclust* function find the optimal number of clusters according to the *silhouette* criteria considering normalised numerical variables.
b. Compute a 3-means clustering on the *framingham* dataset restricted to normalised numerical variables with 25 initialisations and plot the resulting clustering with the *fviz_cluster* function.
c. In a single plot-window, plot 2,3,4 and 5-means clustering of the same data (use the *grid.arrange* function of the *gridExtra* package.
d. Compute the distance matrix between all examples considering the normalised numerical variables with the dist function.
e. Plot the corresponding dendrogram.
f. Add colored bloc arround clusters. And plot the resulting clusters with the *fviz_cluster* function.

7. Classification

a. Define ***y*** and ***x_binary*** as respectively the '*TenYearCHD'* binary labels and the matrix containing all the binary variables of the *framingham* dataset and create a ***preprocessed_dataset*** as the concatenation of ***x***, ***x_binary***, and ***y***.
b. Convert all binary variables to factors.
c. Create ***train*** and ***test*** datasets from the ***preprocessed_dataset*** considering a 80-20% split after examples shuffling.

d. Considering the predictive task of predicting *TenYearCHD'* from all other variable, compute *knn* predictions with the *knn* function of *class* package and the corresponding accuracy. Comment.
e. Train a naive Bayes model on ***train*** that can predict the 'TenYearCHD' variable from all other variables, compute its associated predictions on ***test*** and the corresponding accuracy (package *e1071*, function *naiveBayes*).
f. Same thing for a *SVM* model (function svm).
g. Compute the recall and precision of KNN, naive Bayes and SVM predictions.
h. What is the predictive power of the following task : predict *currentSmoker* from *male, diabetes, age, education, totChol, BMI, glucose* with a SVM model ?

8. Regression

a. Train a SVM model on ***train*** that can predict the *glucose* variable from all other variables, compute its associated predictions on ***test*** and the corresponding RMSE.
b. Compute *KNN* predictions of the *glucose* variable from all other variables on ***test*** and the corresponding RMSE (package *FNN,* function *knn.reg)*.
c. Compute a model comparison pipeline on the ***preprocessed_dataset*** considering a **dummy** regressor predicting the average label value of the train data, a **KNN** regression model, a **SVM**. For the evaluation protocole considered a **10-fold cross validation** (package *caret*, function *createFolds*) **repeated 30 times**, compute the average **RMSE** and plot the corresponding boxplots.