Labwork on **Machine learning initiation**
DATA SCIENCE

Nicolas Sutton-Charani
Euromov - DHM

Solutions will be given on this online notebook.

Exercice 1: K-means clustering

Aim = targeting teenager market segment for efficient advertisement

Juhi Garg

Now a days, every age group is actively using social networking sites. But the most popular segemnt on social networking sites is the group of teen agers. They not only share their every moment of life on these sites but also showcase their interests in several other fiels like sports, music, clothings etc. Social networking websites could be very useful for online retailers to capture such interests of people. They can get insights from these data and can target the segments together which have common interests to advertise their products. For instance: If they can find out patterns based on genger, age etc that these age-group of people are highly interested in football, so why not just showing these group of people the footbal accessories to efficiently convert them to their customers instead of showing to entire gropu of people which might not be interested in football. K-mean clustering provides an efficient way of clustering segments together which have the similar characteristics. The below analysis shows how it can be used to segment teenagers based on their interest and then target only those segments for advertisisng the products.

About the dataset used: For this analysis, I have used a dataset representing a random sample of 30,000 U.S. high school students who had profiles on a well-known SNS. To protect the users' anonymity, the SNS will remain unnamed.

The dataset contatins 40 variables like: gender, age, friends, basketball, football, soccer, softball, volleyball,swimming, cute, sports, rock, god, church, hair, mall, clothes, hollister, drugs etc whcih shows their interests

1. Import the *teens* dataset from the following link: here.
2. Preprocessings
   (a) Compute the dimensions of the *teens* dataset.
   (b) Return the number of missing data.
   (c) Filter *teens* on the complete examples with the *na.omit* function.
   (d) Convert the *gender* variable into a numerical one.
   (e) Normalise the *teens* dataset with the *scale* function.
3. Data description: Principal Component Analysis (*PCA*)
   (a) With the *PCA* function of the *FactoMineR* package compute a PCA and print the resulting eigen values.
   (b) With the *fviz_* functions, plot the the percentage of explained variance per dimension and the 2 PCA plots (1 for examples and 1 for variables).

   (c) Plot the contributions of each variable to the 5 first dimensions with the *get_pca_var* and *corrplot* functions of the *corrplot* package.

4. Clustering

   (a) Compute a 2-means clustering on the preprocessed *teens* dataset with 25 different initialisations and plot the resulting clustering with the *fviz_cluster* function.
   (b) In a single plot-window, plot 2,3,4 and 5-means clustering of the *teens* dataset (use the *grid.arrange* function of the *gridExtra* package.

5. Optimisation: with the *fviz_nbclust* function fid the optimal number of clusters according to the 3 available criteria.

Exercice 2: Hierarchical clustering

1. Import the native dataset *USArrests*
2. Clustering
   (a) Compute the distance matrix between all examples with the *dist* function.
   (b) Compute a hierarchical clustering of the data with the *hclust* function.
   (c) Plot the resulting dendrogram.
   (d) Compute the "agglomerative coefficient" (quality index) of the resulting clustering.
3. Optimisation
   (a) With the *agnes* function, compute the agglomerative coefficient of the following methods: "average", "single", "complete", "ward".
   (b) Compute the best hierarchical clustering according to the previous question.
   (c) With the *cutree* and *rect.hclust* functions, cut the dendrogram of the previous question in the optimal number of clusters and assign clusters to each examples.
   (d) Plot the corresponding clustering with the *fviz_cluster* function.
   (e) With the *dendextend*, compare 2 different hierarchical clusterings in a single plot.

Exercice 3: Classification

1. Import the dataset contained at the following url: "https://stats.idre.ucla.edu/stat/data/binary.csv"
2. Preprocessings: convert the *admit* variables into factor.
3. **Learning**
   (a) Randomly shuffle the dataset rows and create a training set corresponding to $\frac{3}{4}$ of the dataset and a testing set with the remaining data.
   (b) Learn a logistic regression classifier on the training set.
   (c) With the classifier learnt at the previous question make predictions on the testing set.
4. Evaluation:
   (a) Compare the obtained predictions with the real labels of the testing set and print the corresponding '*table*' object.
   (b) Compute the corresponding accuracy, precision and recall and print them.
5. Models comparison
   (a) Compute a '*baseline*' classifier which systematicaly predicts the majority label and evaluate on the same data as in questions 3 and 4.

(b) Set up a pipeline in order to compare the accuracy of the baseline classifier, a logistic regression model, a KNN classifier (with $K = 20$) and a SVM. The same evaluation protocole (shuffle, then $\frac{3}{4} \rightarrow$ train set and $\frac{1}{4} \rightarrow$ test set) has to be repeated several times.

(c) Create a data frame containing all the accuracies of the different computed classifiers and print the mean accuracies of each models with 3 digits.

(d) Plot the results with 1 boxplot per model with the native '*boxplot*' function and then draw a more elaborated plot with ggplot.

6. Scale all attributes in order to have values in [0, 1] and re-do the previous question.

7. Optimisation (*gridsearch* approach): find the optimal value of the '*cost*' SVM hyperparameter with a 10-fold cross-validation procedure applied to the whole dataset with a performance comparison between SVM models trained with the foloowing values: 0.5, 0.6, 0.7, 0.8, 0.9 and 1.