**Data Science - complete exercise**

In this notebook you are asked to develop a full R project.

The standard steps of a Data Science projects include, among others:

- Acquiring data

- Exploring and cleaning data

- Set up our modeling objective

- Fit several models and select the best one

- Make a report explaining your process and the expected results

- Put our model into production

- Apply our model to new data

We also want our project to be a proper Software Engineering project, so it should contain:

- Train and Test separately

- Logging

- Unit tests

- Control errors

- Configuration in separate folders

In the following sections you will find some hints on how to approach the problem

# Setup the environment - main & logging [15 min]

Exercise:

* Create a main.R file that will be run in order to execute the complete process
* The file is run with source(‘main.R’)
* This file will include you “import” statements (library in R)
* Initialize the logger handler and the logging level
* Test your setup:
  + create a new file (get\_data.R)
  + Includes print(‘Hello world’)
  + Includes a logging sentence
  + Run get\_data.R from main
  + Execute source(‘main.R’) in the terminal and observe the result

Bonus:

* Create the logger file name including a timestamp so that each time you run the process a new log file is created

Deliverable:

* script `main.R` to be run by the instructor
* file with logs written

# Acquiring data [30 min]

In this step you will get your data from an external source. You should think as you have to run this process several times (ex: each day). You should get your data, process it (if need it) and store it locally so you can start doing something with it.

Think that this is your basic dataset, so you should check it is correct for the purpose of your task and ensure it does have the format you need.

**Exercise:**

* get a dataset to work with (scrapping webpage, downloading dataset...).
  + If you don’t find a proper dataset, ask your instructor
* place all data acquisition process in a R script that can be run completely
  + The file should be named get\_data.R
  + You can run it with source(‘get\_data.R’)
  + Besides, you should place it into main.py as a first step, and run the whole process by running main.
* Include logging after several steps
* Include some assertions after several steps
* At the end, it should save (formatted) data into a file
* Place the name of the output file and the logging in a variable in a separate file (config.R); and load this configuration file at the beginning of your main script (source(config.R))

Deliverable:

* script `get\_data.R` to be run by the instructor
* file with logs
* file with downloaded data formatted

# Exploring and cleaning data [45 min]

Examples:

- remove NA

- convert to proper types

- remove outliers...

- filter only important rows...

- plot some statistics, distribution...

**Exercise:**

* Create a .Rmd file that explores the data.
* It should contains, at least:
  + number of rows and columns
  + distribution of continuous variables
  + frequency of categorical variables
  + NAs by column
* Checking some conditions on data (ex: id variables not to have NAs, only positive columns...)
* Data should be read from the file that you have created in previous exercise. File name has to be loaded from variable in a config file.

Deliverable:

* Rmd file to be run completely by the instructor; html generated

# Set up our modeling objective [1 h]

- define our objective

- build the target

- select the features

- build our training dataset

- split train, test, validation

- think of the metric you want to optimize

**Exercise:**

* You should create a .Rmd file for exploring your modelling stage (`explore\_training.Rmd`)
  + Read original data
  + Preprocess data
  + Split data train/test, preprocess, create class
  + Define your model and your experiment setting (cv, metric)
  + Fit the model with data
  + Measure the performance on test data
  + Plot variable importance
* Include some assertions (ex: at least N rows, result metric above/below a threshold,...)
* Include logging after some steps

Notes:

Explore [caret package](http://caret.r-forge.r-project.org) and some of its functions.

# Fit several models and select the best one [30 min]

Now you have your training pipeline clear, you should put the necessary code into a R script that will be run automatically each time.

- apply different models with different set of parameters

- report metrics on each one and select the final model

\*Hint\*: You can use `tuneGrid` parameter in the `train` method of the `caret` library. Example from https://topepo.github.io/caret/model-training-and-tuning.html

gbmGrid <- expand.grid(interaction.depth = c(1, 5, 9),

n.trees = (1:30)\*50,

shrinkage = 0.1,

n.minobsinnode = 20)

nrow(gbmGrid)

set.seed(825)

gbmFit2 <- train(Class ~ ., data = training,

method = "gbm",

trControl = fitControl,

verbose = FALSE,

## Now specify the exact models

## to evaluate:

tuneGrid = gbmGrid)

gbmFit2

**Exercise:**

* - create a .R file (`fine\_tune.R`) where you apply the complete training process
* - it should include fine-tunning and testing different models and select the best one

# Make a report explaining your process and the expected results [optional, 30 min]

- present results to your manager / teacher / ...

**Exercise:**

* - You should create a summary notebook explaining the process in order to convince your manager / teacher / inverstors that your work is solid and trustable
* - Includes: some data exploration (just the highlight)

# Put our model into production [30 min]

**Exercise:**

* - Create a file with the final code (`train.R`) that will be used for training:
  + - Read best parameters from config file
  + - Retrain your selected model on whole dataset (not only train)
  + - Save trained model into a file
* - Create a `predict` function in a new file (`predict.R`) to be applied on new data:
  + - You have to read saved model and apply it to a new data
* - Apply the function to some examples (create a file for that)