jn_xgboost

The XGBoost architecture is intended to make testing different XGBoost hyperparameters as easy as possible. It is divided into several scripts. The main user-facing one is a setup script; its default name is script is **jn_xgboost_setup.R**. It is used to set file paths, hyperparameters to test, number of cores etc. and initiates the modeling process by sourcing the second script, **jn_xgboost_hub.R**. It's called a hub script because it sources other scripts depending on the settings detected in the setup script. Currently, it sources a **run** and **eval** script depending on whether or not regression or softprob classification is selected by the user. The **run** scripts train one model each for all the possible combinations of the hyperparameters selected in the **setup** script, while the **eval** script generates plots depending on whether regression or classification were selected and whether the run is a cross-validation one or uses specific validation data not used for the training process. Lastly, **jn_xgboost_hub.R** sources **jn_xgboost_report.Rmd**, which prints an html file listing all XGBoost models trained, their hyperparameters and their training and validation or cross-validation accuracy.

As mentioned, the **setup** script is the main, and usually only one a user needs to look at and edit. It is also the only one not explicitly assumed to be in the same folder as the other scripts, meaning the user can and should make different **setup** scripts with different names for different runs. Inputs are assumed to be split into different rds files containing predictor and outcome variable data.

Things to specify in the setup script – everything is detailed by comments in the script:
1. File paths: Path to jn_xgboost_hub.R. All inputs - training data and, optionally, validation data. Output folder.
2. Model hyperparameters.
3. Miscellaneous: Run name, Number of cores.

Outputs:
1. A report (html file) showing run details and a list of the trained models sorted by performance.
2. All trained models stored as rds files.
3. Performance plots for each model. Type of plots created depends on whether or not a cross-validation run or a regular run was selected.

Note: The subfolder **jn_xgboost_demo_data** contains training and validation data that can be used for a demo run of the scripts. The default **jn_xgboost_setup.R** is mostly set up for this already, only requiring to set the path to the scripts.