Assignment III: Functional Programming, Machine Learning, and Dashboards

Data Science Project Management (DS400) | Winter Term 2024/25

Formal requirements

Submit your assignment earlier than **January 07, 2025, 12:00 noon**. The submission must include your code, results, and explanations for code and results, for both **R** and **Python**.

Submit your assignment via ILIAS. Upload the `*.html`-files that were rendered from your `*.Rmd` and `*.ipynb`, or `.qmd` solutions. You can optionally upload the `*.Rmd` and `*.ipynb`, or `qmd` files as well. You must use the following naming convention: `<Lastname>_<Firstname>_Assignment##_<Language>.<extension>` (e.g., `Lovelace_Ada_Assignment03_R.html`).

For the Dashboard, submit a separate Python (`.py`) and R script (`.R`) as well as all files needed to run the dashboard independently of your other scripts. Use the following naming convention for the dashboard: `<Lastname>_<Firstname>_Assignment##_Dashboard.<extension>` (e.g., `Lovelace_Ada_Assignment03_Dashboard.R`).

You will receive up to **7.5 points** upon passing this assignment.

Code of conduct

By submitting the assignment, you are acknowledging that the submitted assignment is your own work. This implies that …

- the work you submit is your own
- the code you wrote is your own
- any comment or interpretation is in your own words
- you made clear whom you worked with

Disregarding this code of conduct will result in failure of the assignment!

Functional Programming

Part 1: Function `describe_data()`

1. [R] Program a function that describes a dataset. The function should be called `describe_data()` and take an R dataframe as input. The function should not return anything but instead print the following information:

    - number of rows and columns
    - a section for all categorical variables (i.e., factor variables)
    - a section for all continuous variables (i.e., numeric and integer variables)
    - a section for all other variables (i.e., character variables, logical variables, etc.)
    - for categorical variables, absolute and relative frequencies (in %) – including `NA` as a category if there are any – should be printed with one line per a variable's category
    - for continuous variables, a table should be printed with variables as rows and `N`, `min`, `max`, `median`, `mean`, standard deviation (`SD`), and `NA` as columns
    - for all other variables, the number of missing values should be printed.

    Apply your function to the artificial dataset `artificial_data.csv` provided in `data.zip`. The result should look like the output shown below.

```
Number of rows and columns:
==============================
Rows: 11
Columns: 5


Categorical variables:
==============================
Variable: gender
------------------------------
female: 6 (66.67%)
male: 3 (33.33%)
NA: 2 (18.18%)

Variable: adult
------------------------------
yes: 8 (72.73%)
no: 3 (27.27%)


Continuous variables:
==============================
        N min max median     mean       sd Missings
verbal  9   2   8      4 4.555556 2.068279        2
math   11   1  10      6 6.000000 3.286335        0

Other variables:
==============================
comments: 5 NA
```

Wine Quality

Next, you will work with the `winequality-white.csv` dataset. The dataset includes wine quality ratings and values on several physicochemical tests for vinho verde wine samples from Portugal (see Cortez et al., 2009). The original information file is included in `data.zip`.

The dataset has 4898 observations and 12 columns, including the target variable `quality`. The goal is to predict the quality of the wine based on the other variables.

Part II: Data Inspection

2. Load the dataset and inspect the first few rows. What are the column names and their data types? Are there any missing values? If so, how many are there in each column?
3. Create a plot that shows the distribution of the target variable `quality`. Include the number of cases for each quality rating at the top of each bar. What can you infer from the plot?
4. The dataset's info file contains a note saying that "several of the attributes may be correlated." Create a correlogram to visualize the correlation between the variables. What can you infer from the correlogram?

Part III: Data Preprocessing

Using the wine attributes, we want to predict the quality of the wine as quality categories. To do so, we first need to preprocess the data.

5. Create a new column `quality_cat` that categorizes the quality of the wine into four categories: `low`, `medium`, `high`, and `prestigious`. The categories should be based on the following criteria: `low` for

quality ratings below 5, `medium` for quality ratings of 5 and 6, `high` for quality ratings of 7 and 8, and `prestigious` for quality ratings above 8.

6. Split the dataset into a training and a test set. Use 70% of the data for training and 30% for testing. To make sure your results are reproducible, use `42` as seed.

Part IV: Machine Learning

7. Explain why logistic a regression is not a suitable model for this dataset.
8. Train a K-Nearest Neighbors model to predict the quality of the wine. You may need to prepare the data for use with the model. Make sure your training process is reproducible.
9. Train a random forest with 10 trees to predict the quality of the wine. You may need to prepare the data for use with the model. Make sure your training process is reproducible. In R, change the `engine` to `"ranger"` (ranger is a fast implementation of random forests for high dimensional data).
10. Evaluate the performance of the models using the test set. Inspect the confusion matrix and report the accuracy of the models. Do you see any problems with this classification?
11. What would be a naïve baseline for this prediction task? Are your models better than this naïve baseline?
12. There is a second dataset for red wine quality ratings, `winequality-red.csv`. Use your best model to predict the quality of the red wine. You may need to prepare the data for this prediction. Inspect the confusion matrix. What is the accuracy of the model on the red wine dataset? Is it better than a naïve baseline?

Part V: Dashboard

Now we want to make this type of data analysis accessible to non-technical users. The best way is to build a dashboard which allows users to interact with the data and the models. To make the task easier, we'll focus only on the red wine dataset.

13. Built a dashboard that allows users to select a model (KNN or RF) and the variables to be used for the prediction of the wine quality category. The dashboard should be built using the `shiny` library in R and python. The dashboard should include the following elements:

    - A title explaining the purpose of the dashboard.
    - A field to select variables to be used for the prediction.
    - A field to select a model (KNN or RF).
    - A field to set the hyperparameters of the selected model.
    - A field to set the training/test split ratio.
    - A field to set a seed for reproducibility.
    - A correlogram of the selected variables.
    - A button to run the prediction.
    - A bar plot of the predicted wine quality categories.
    - The performance of the model (accuracy).

    Regarding the layout you are free to choose. Just make sure it is intuitive and easily readable. The best three dashboards will receive an extra point.

You may choose to build a "standard" dashboard in one language and a "best-version" dashboard in the other language. However, both dashboards should be fully functional and include all the elements mentioned above.

Remember to submit the dashboard as a separate R and Python script than run independently of your other scripts.