Cross-validation and Hyperparameter optimization

Part 1: Pipelines

see Ch. 05 in book “Python Machine Learning” by Raschka & Mirjalili

Chapter 6 Lecture Overview

- Monday (18/03): Pipelines
- Thursday (21/03): Cross-validation and Hyperparameter optimization
- Easter holiday
- Thursday (04/04): Evaluation metrics

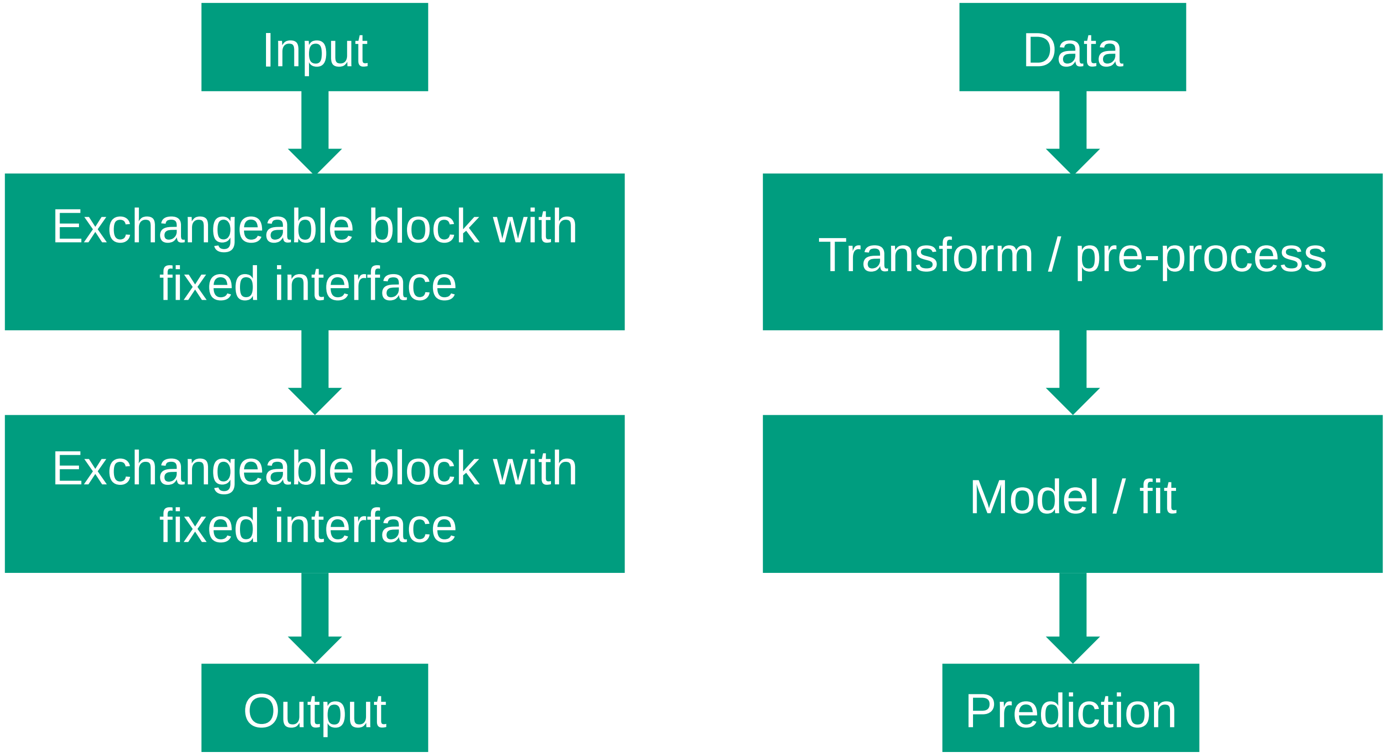
Chapter 6 Content Overview

- Today
  - Pipelines
- Thursday this week
  - Validation / Cross-validation
  - Learning and interpretation of validation curves
  - Grid search and random search for selecting good hyperparameters
- Thursday after easter
  - Confusion matrix (one-versus-all and one-versus-one)
  - Receiver Operator Curve (ROC) and Area Under the Curve (AUC)
  - Metrics for an unbalanced dataset
  - Multiclass metrics

What is a pipeline?

- Any suggestions?
- Pipelines are a **tool for chaining** multiple data processing steps and ML models into a single object
- They are useful for **encapsulating all** the preprocessing steps (data scaling, feature selection, feature engineering, etc.) and the ML model **into one entity**

What is a pipeline?


Input
Exchangeable block with fixed interface
Exchangeable block with fixed interface
Output
Data
Transform / pre-process
Model / fit
Prediction

Key benefits of pipelines

- **Sequencing**: Pipelines allow for defining a sequence of data processing steps and ML algorithms
- **Consistency**: Pipelines ensure that all preprocessing steps are applied consistently
- **Convenience**: Pipelines provide a convenient way to fit, predict, and evaluate models with a single call
- **Hyperparameter tuning**: They are especially convenient when we will be looking at tuning of hyperparameters in the next lecture.

Pipelines - Example

`Pipeline_with_LDA_and_logistic_regression.ipynb`

Hyperparameters and data - split

- What are **hyperparameters**?
  - They are **non-trainable** model-parameters that affect performance
  - E.g Max depth in a decision tree, or number of trees in a random forest
- Hyperparameter-tuning is about looking for the models **optimal set of hyperparameters**
- Up until now we have split datasets into two partitions: “train” and “test”.
- When working with hyperparameter tuning we will introduce a third partition, “validation”
  - Do you have any suggestions as to **why** we would want to split a dataset into ”train”, “validation”, and “test”?

Thank you for listening