DS-210: Programming for Data Science

Lecture 12: Cross–validation. Hyperparameter tuning.

Midterm

- **Date and time:** Monday 2/28, 12:20–1:10pm
- **Location:** here (MCS B37)

- Open book
- No electronic devices

Midterm

- **Date and time:** Monday 2/28, 12:20–1:10pm
- **Location:** here (MCS B37)

- Open book
- No electronic devices

Content:

- data analysis in Python
- general data analysis concepts
- Python features explained in class

Format:

- answer quiz questions
- explain simple concepts
- write simple code

Terminology

Parameters

- Variables fixed in a specific instantiation of a model
- Examples:
  - coefficients in linear regression
  - decision tree structure and thresholds
  - weights and thresholds in a neural network

Hyperparameters

- Also parameters, but higher level
- Examples:
  - number of leafs in a decision tree
  - number of layers and their structure in a neural network
  - degree of a polynomial

Terminology

Parameters

- Variables fixed in a specific instantiation of a model
- Examples:
  - coefficients in linear regression
  - decision tree structure and thresholds
  - weights and thresholds in a neural network

Hyperparameters

- Also parameters, but higher level
- Examples:
  - number of leafs in a decision tree
  - number of layers and their structure in a neural network
  - degree of a polynomial

Hyperparameter tuning

- Adjusting hyperparameters before training the final model

Model selection

- Deciding on the type of model to be used (linear regression? decision trees? ...)

Challenges

Big goal: train a model that can be used for predicting

Intermediate goal: select the right model and hyperparameters

Challenges

Big goal: train a model that can be used for predicting

Intermediate goal: select the right model and hyperparameters

How about trying various options and seeing how they perform on the test set?

Challenges

Big goal: train a model that can be used for predicting

Intermediate goal: select the right model and hyperparameters

How about trying various options and seeing how they perform on the test set?

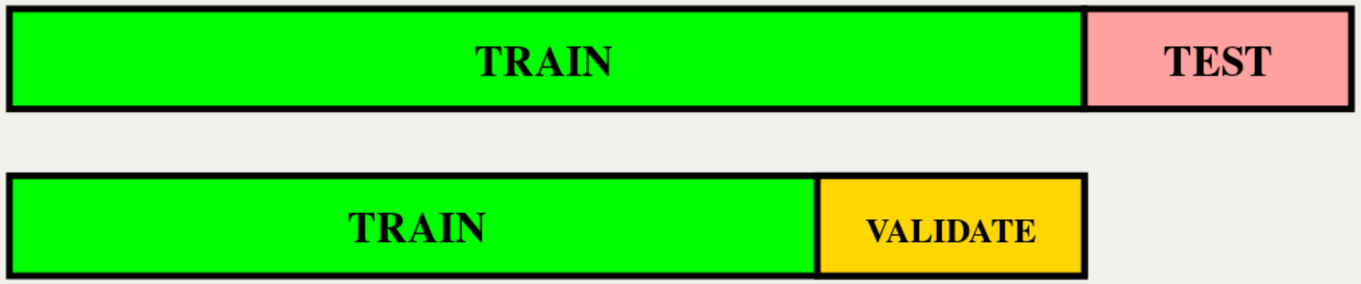
Information leak danger!

- If we do it adaptively, information from the test set could affect the model selection

Cross–validation attempts to solve this problem

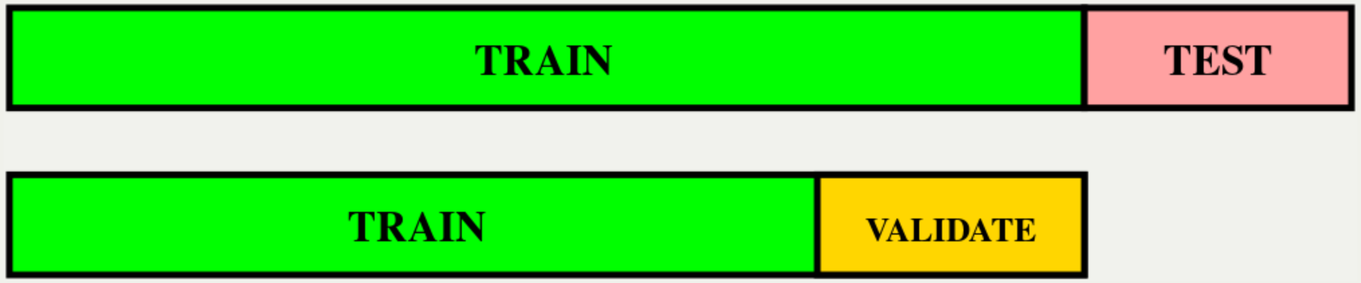
Holdout method

- Partition the training data again: training set and validation set
- Use the validation part to estimate accuracy whenever needed



Holdout method

- Partition the training data again: training set and validation set
- Use the validation part to estimate accuracy whenever needed



Pros:

- Very efficient
- Fine with lots of data when losing a fraction is not a problem

Cons:

- Yet another part of data not used for training
- Problematic when the data set is small
- Testing part could contain important information

k–fold cross–validation

- Partition the training set into k folds at random
- Repeat k times:
    - train on $k - 1$ folds
    - estimate the accuracy on the k-th fold
- Return the mean

k–fold cross–validation

- Partition the training set into k folds at random
- Repeat k times:
  - train on $k - 1$ folds
  - estimate the accuracy on the k-th fold
- Return the mean

Pros:

- Every data point used for training most of the time
- Less variance in the estimate

Cons:

- k times slower

LOOCV: Leave–one–out cross–validation

- Extreme case of the previous approach: separate fold for each data point
- For each data point q:
  - train on data without q
  - estimate the accuracy on q
- Return the mean of accuracies

Cons:

- Even more expensive

Many other options

- Generalization: leave–p–out cross–validation enumerates over $\binom{n}{p}$ subsets
- Sampling instead of trying all options
- A variation that ensures that all classes evenly distributed in folds
- ...

(scikit-learn docs are pretty good: https://scikit-learn.org/stable/modules/cross_validation.html)

Many other options

- Generalization: leave–p–out cross–validation enumerates over $\binom{n}{p}$ subsets
- Sampling instead of trying all options
- A variation that ensures that all classes evenly distributed in folds
- ...

(scikit-learn docs are pretty good: https://scikit-learn.org/stable/modules/cross_validation.html)

Hyperparameter searching

Many helpful tools:

https://scikit-learn.org/stable/modules/grid_search.html

Getting ready for the second part of the course

- Text editor
- Rust installation
- Terminal usage
- Version control: git (will be mentioned briefly next time)