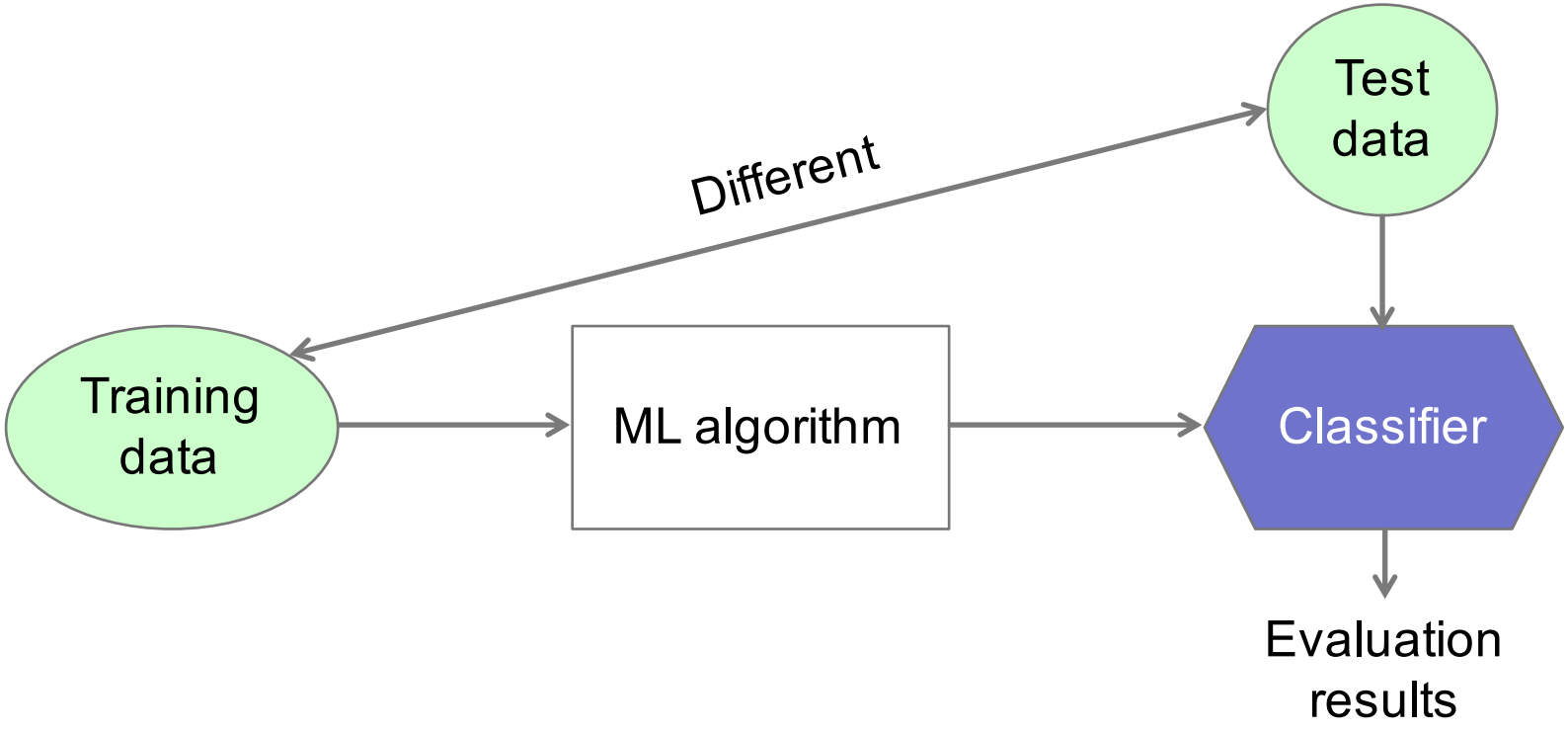
Training and Testing

Nikola Milikić
nikola.milikic@fon.bg.ac.rs

Jelena Jovanović
jeljov@fon.bg.ac.rs

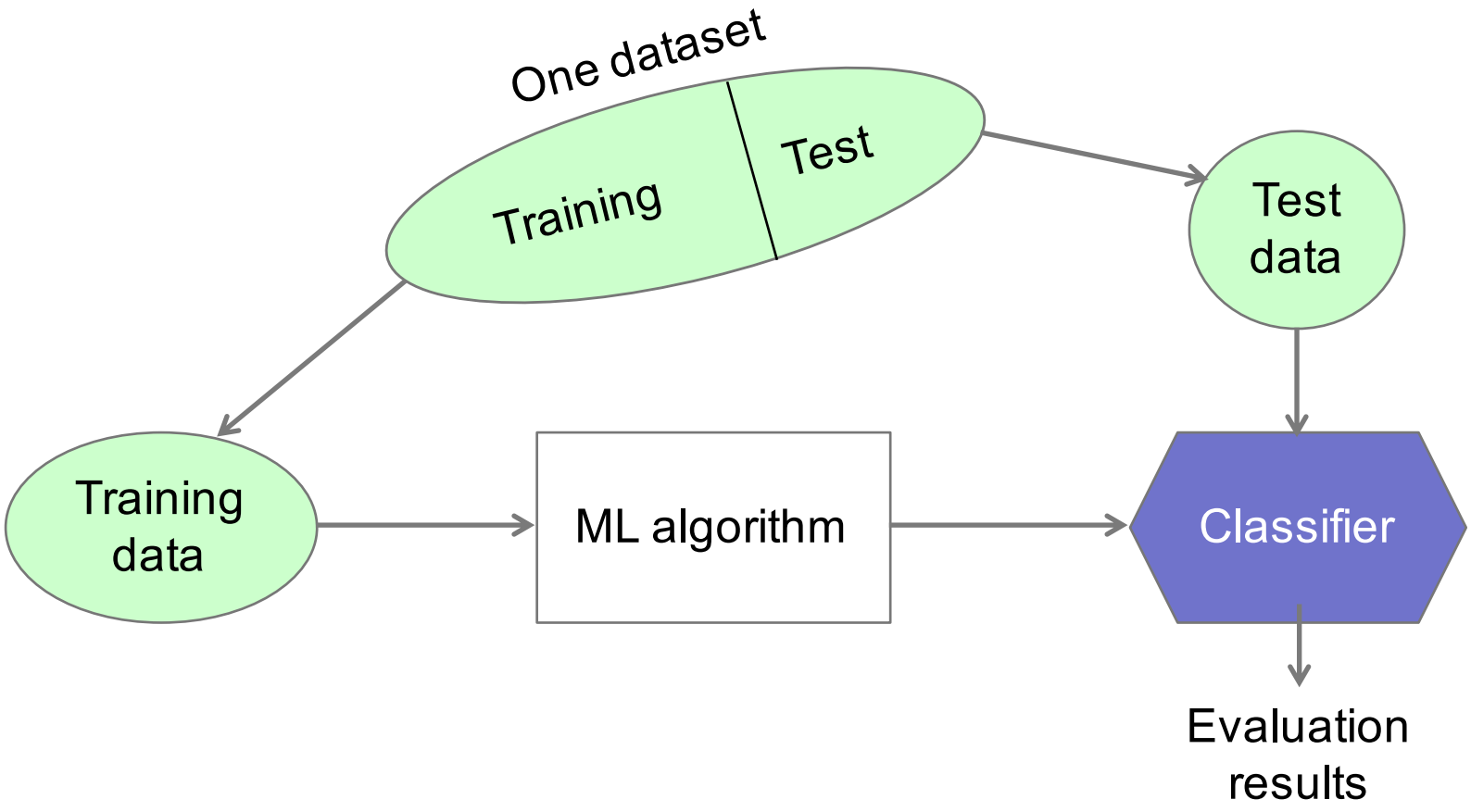
Training and Testing

- Training data is used for building a ML model
- Testing data is used for measuring performance of a ML model
- Training and testing data should be different, mutually independent and created by random sampling



Training and Testing

- In case we have one dataset (for instance, in one file), we need to split the original dataset into subsets for training and testing



Holdout method – different *random seed* values

- Random seed is a number (or a vector) used to initialize a pseudorandom number generator

- Testing J48 classifier results over the dataset *diabetes.arff*
- With *Percentage split* set to 90% for different *random seed* values we get different results:

Random seed	1	2	3	4	5	6	7	8	9	10
Accuracy	0.753	0.779	0.805	0.74	0.714	0.701	0.792	0.714	0.805	0.675

$$\bar{x} = \frac{\sum x_i}{n} = 0.7478 \qquad \sigma^2 = \frac{\sum (x_i - \bar{x})^2}{n-1} \qquad \sigma = 0.046$$

Cross-validation

- The goals of cross-validation are:
  - to make the most efficient use of the available data
  - to overcome the problem of overfitting, thus making the predictions more general

Cross-validation

Consists of the following steps:

- Splitting the original dataset into *k* equal parts (folds)
- Taking one fold aside for testing, and using the rest (k-1) folds for training; after training is done, the testing fold is used for measuring the performance of the built model
- Repeating the process *k* times, by setting aside a different testing fold each time

10-fold cross-validation

- k = 10
- Dataset is divided into 10 equal parts (folds)
- One fold is set aside in each iteration (10 iterations in total)
- Each fold is used once for testing, nine times for training
- The overall score is the average of the scores obtained in the 10 iterations

10-fold cross-validation

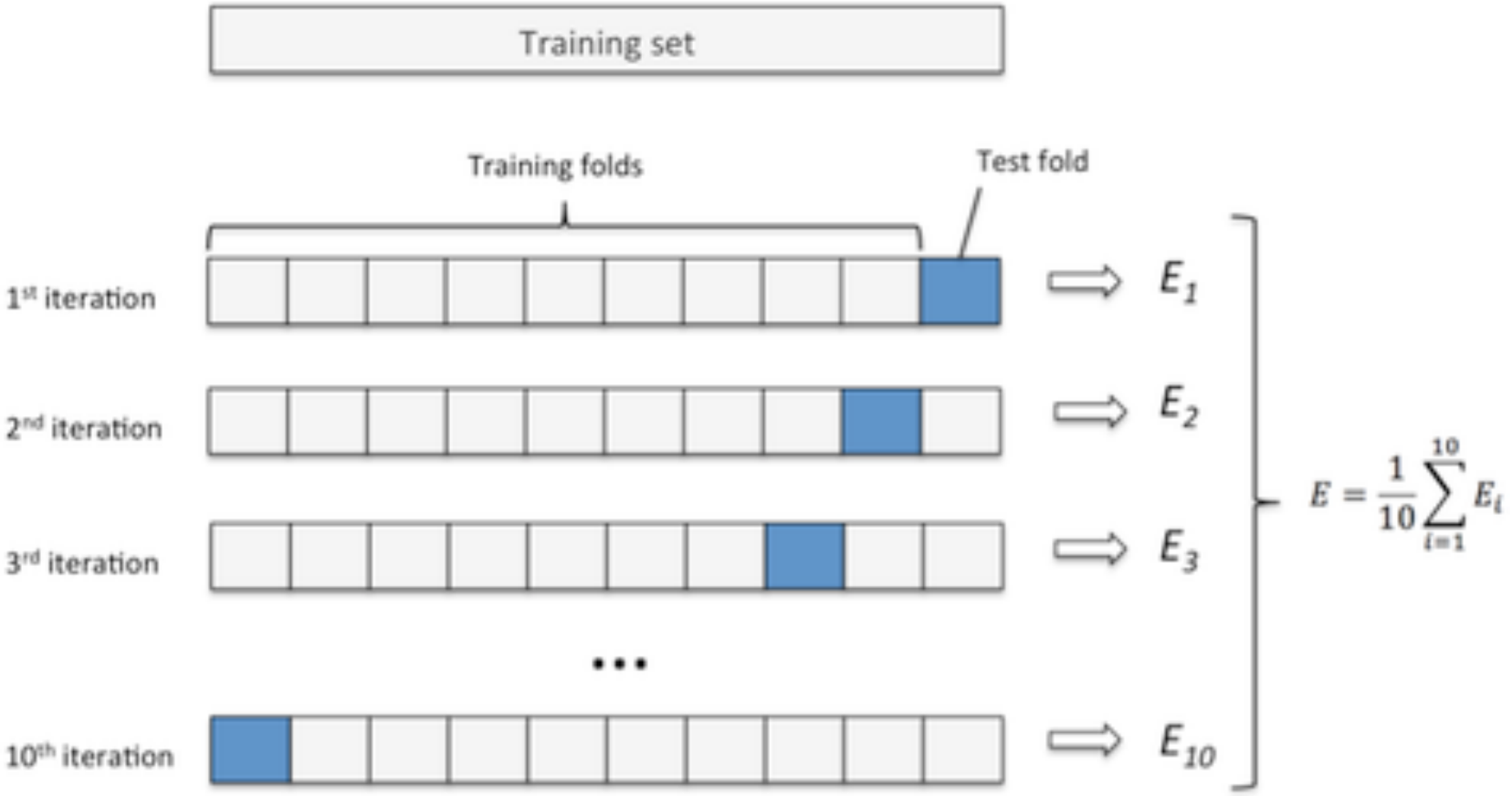


Image source: http://sebastianraschka.com/faq/docs/evaluate-a-model.html

Stratified cross-validation

Ensures that the distribution of class values in the overall dataset is preserved in all the folds

For example, in the *diabetes.arff* dataset, we have the following distribution of the class values:

- Positive class: 268 instances; 34.9%
- Negative class: 500 instances; 65.1%

Stratification will assure that in each fold we also have (roughly) 34.9% of positive class instances and 65.1% of negative class instances

Cross validation- different *random seed*

- Testing results of J48 classifier over dataset *diabetes.arff*
- With *Cross-validation* set to *10 folds* for different *random seed* values we get different results:

Random seed	1	2	3	4	5	6	7	8	9	10
Accuracy	0.738	0.75	0.755	0.755	0.743	0.756	0.736	0.74	0.745	0.73

$$\bar{x} = 0.7448 \qquad \sigma = 0.0008$$

Smaller deviation and variance with cross-validation

- Previous results for holdout method:

$$\bar{x} = 0.7478 \qquad \sigma = 0.046$$

Recommendations and credits

"Data Mining with Weka" and "More Data Mining with Weka": MOOCs from the University of Waikato. A self-paced session of "Data Mining with Weka" runs until 23 October June 2015.

- Link: https://www.youtube.com/user/WekaMOOC/

(Anonymous) survey for your comments and suggestions:
http://goo.gl/cqdp3l