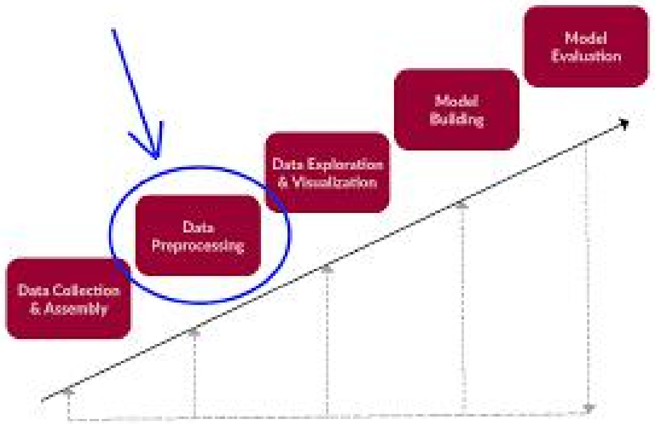
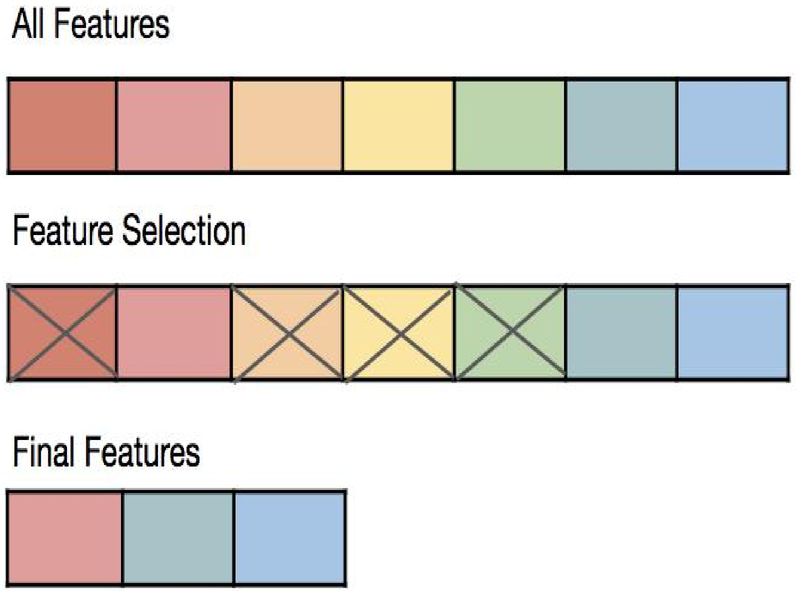
Chapter-2

Data Preprocessing





https://machinelearningmastery.com

Why Data Preprocessing?

- No quality data, no **quality results!**

- Quality decisions must be based on quality data

- ML algorithms required data at high quality

Measures of Data Quality:**Why Data Preprocessing?**

- **Accuracy**: How well does a piece of information reflect reality? [correct/wrong]
- **Completeness**: Does it fulfill your expectations of what's comprehensive? [recored/not]
- **Consistency**: Does information stored in one place match relevant data stored elsewhere?
- **Timeliness**: Is your information available when you need it?
- **Validity**: Is information in a specific format, does it follow business rules?
- **Uniqueness**: Is this the only instance in which this information appears in the dataset?

Why Data Preprocessing?

- Data in the real world is full of dirty:
  - **incomplete**: lacking attribute values
  - **noisy**: containing errors or outliers that deviate from the expected
  - **inconsistent**: lack of compatibility (e.g Some attributes representing a given concept may have different names in different databases)
- To minimize such problems, employ data cleaning routines.
- Before starting data preprocessing, it will be adviceable to have **overall picture** of the data at high level summary such as
  - General property of the data
  - Which data values should be considered as noise or outliers
- This can be done with the help of **descriptive data summarization**

Descriptive data summarization

- Descriptive summary about data can be generated with the help of measure of central tendency of the data and dispersion of the data
- Measure of ***central tendency [computing a typical score on the variable] and it includes***
  - Mean
  - Median
  - Mode
  - Mid-Range
- Measure of ***dispersion[computing the degree to which data is distributed around this central tendency]*** includes
  - range
  - Standard deviation

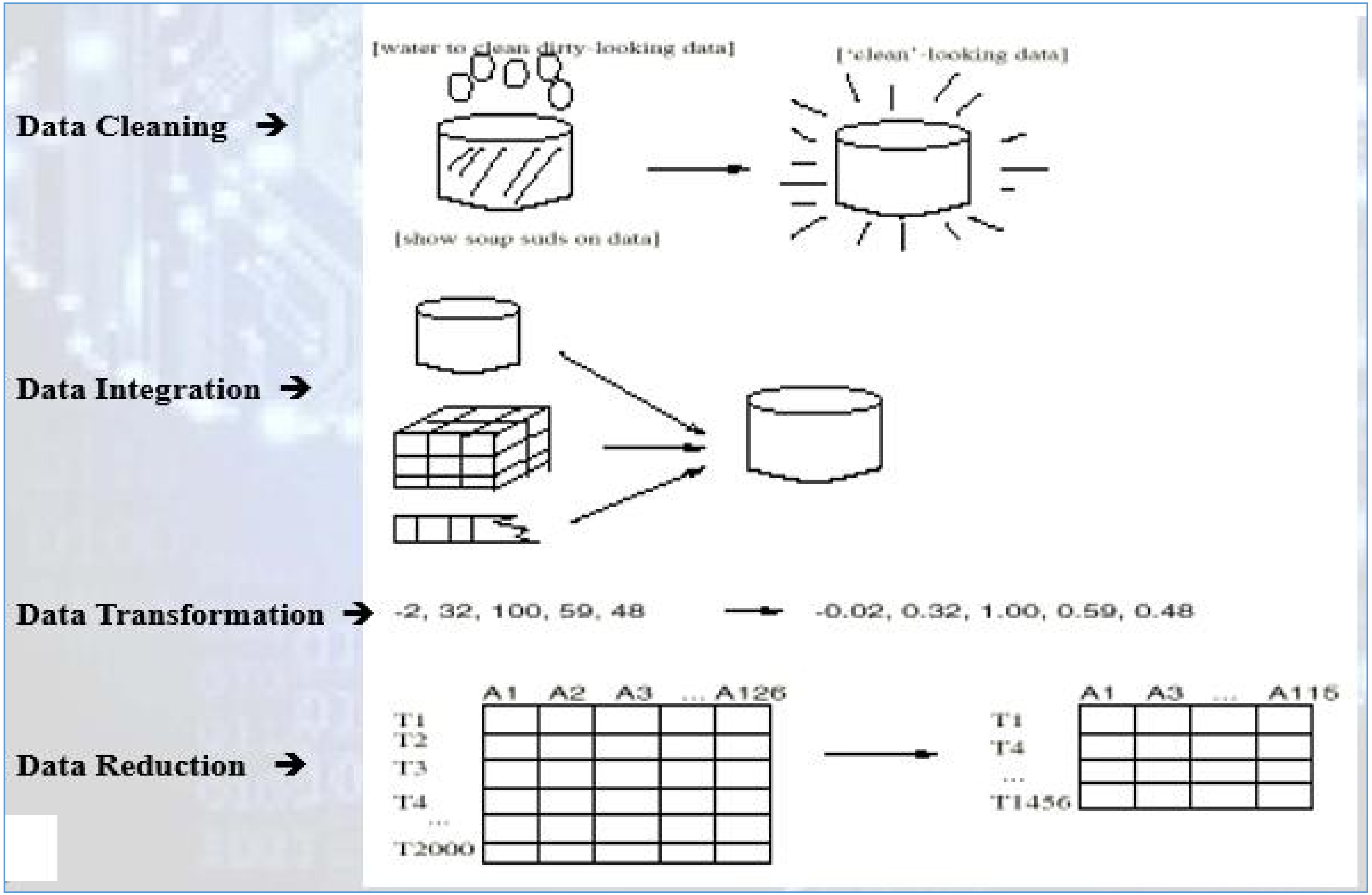
Graphic display of basic descriptive summaries

- Graphical data presentations tools in statistics for the display of data summaries and distributions
  - bar chart,
  - pie chart,
  - line graph
  - Histograms
  - Quantile plot
  - Scatter plot and
  - Loess curves, etc

Major Tasks in Data Preprocessing

- Any activity performed prior to feed to the Learning algorithm is called **pre-processing**
- **Data cleaning**
  - Fill in **missing values**, **smooth noisy data**, identify or **remove outliers**, and **resolve inconsistencies**
- **Data integration**
  - Integration of multiple databases, data cubes, or files (heterogeneous data sources)
- **Data transformation**
  - Normalization and aggregation
- **Data reduction**
  - Obtains reduced representation in volume but produces the same or similar analytical results. Very important for **Big Data** Analysis
- **Data discretization**
  - Data discritization refers to transforming the data set which is usually continuous into discrete interval values.

Forms of Data Preprocessing



9

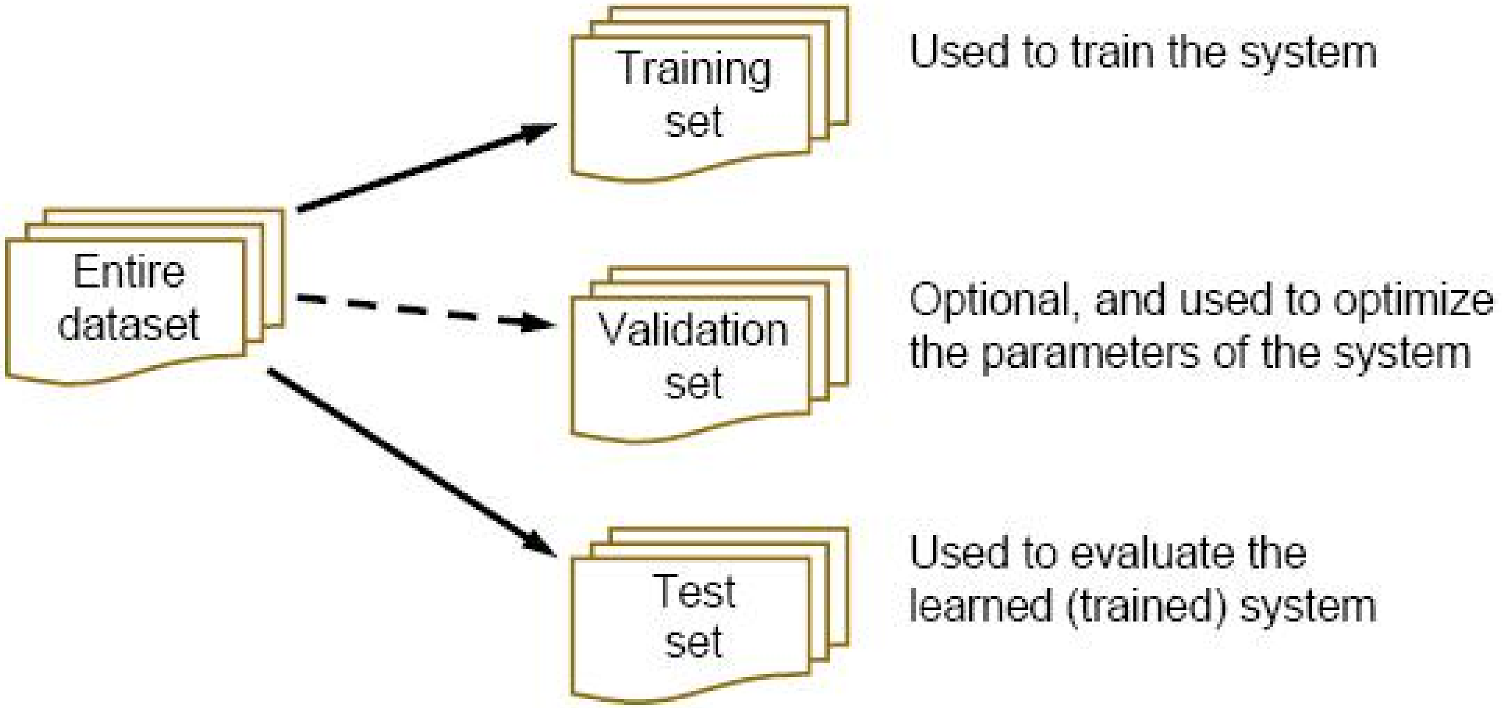
How to Handle Missing Data

- **Ignore the tuple:** usually done when class label is missing
- **Fill in the missing value manually:** tedious and infeasible
- **Use a global constant to fill in the missing value:** E.g., “unknown”, a new class?! Simple but not recommended as this constant may form some interesting pattern and mislead decision process
- **Use the attribute mean:** for all samples belonging to the same class to fill in the missing value with the mean value of attributes
- **Use the most probable value:** fill in the missing values by predicting its value from correlation of the available values
- Except the first two approach, the rest filled values are incorrect and the last two are common.

10

Dataset preparation for Classification

- Proper procedure in some classification system development involves three sets of data :



- Generally, the larger the training data the better the classifier

11

Unbalanced data

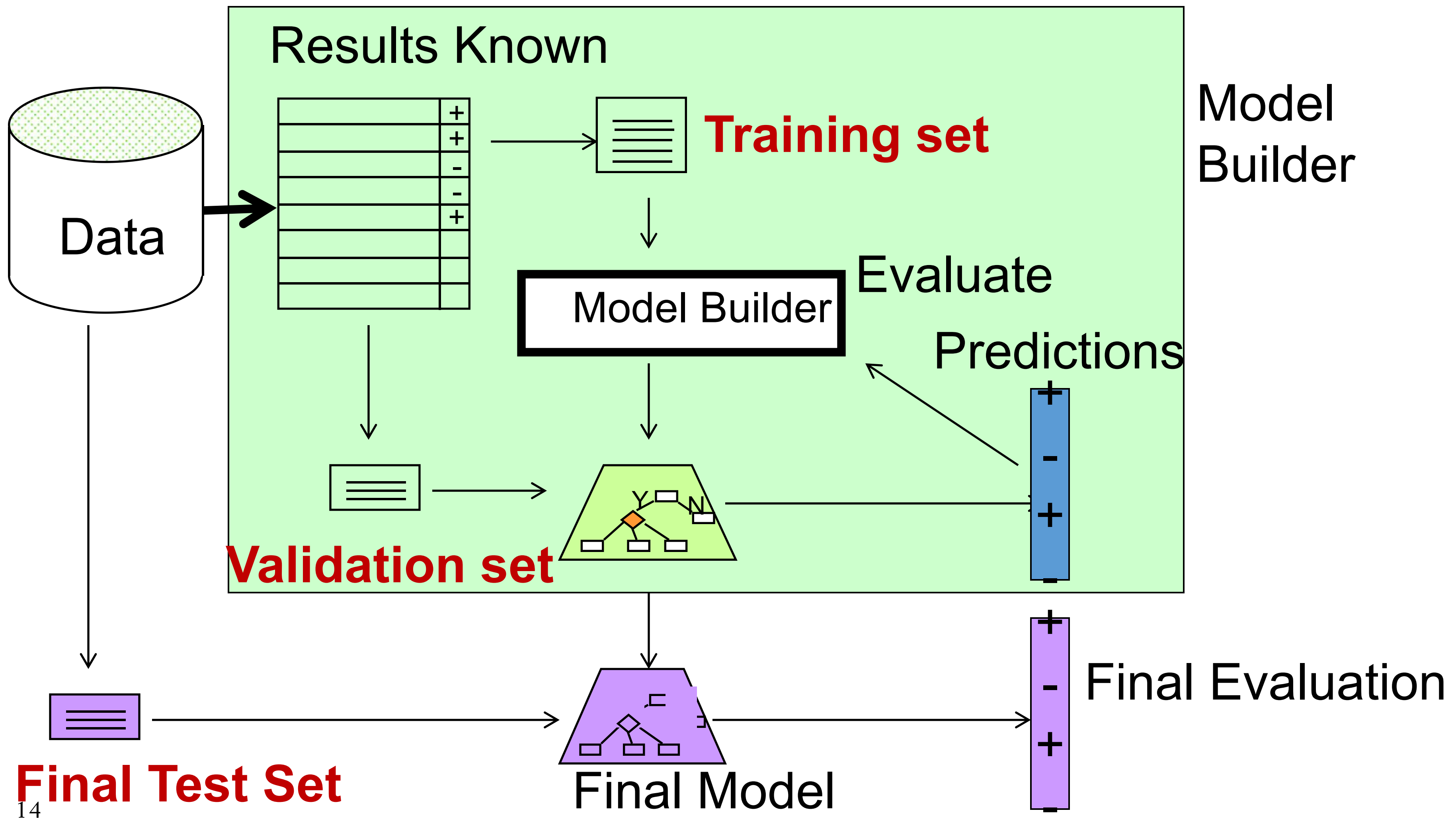
- Sometimes, classes have very unequal frequency
  - medical diagnosis: 90% healthy, 10% disease
  - eCommerce: 99% don't buy, 1% buy
- Majority class classifier can be 97% correct, but useless
- If we have two classes that are very unbalanced, then it will be a bias to evaluate our classifier method
- With two or more classes, a good approach to make a balance between the class instances is to build **BALANCED** train and test sets.

12

Balancing unbalanced data

- With two or more classes, a good approach to make a balance between the class instances is to build **BALANCED** train and test sets
- Approach
  - randomly select desired number of minority class instances
  - add equal number of randomly selected majority class
  - *Stratified sample: advanced version of balancing the data*
    - Make sure that each class is represented with approximately equal proportions in both subsets

Building Classification Model



Building Classification Model: Parameter tuning

- Some learning schemes operate in two stages:
  - Stage 1: builds the basic structure
  - Stage 2: optimizes parameter settings

- Optimizing the parameter setting refers to adjusting important parameters to maximize the performance of the system

- The test data can’t be used for parameter tuning!

Tips: Dataset size

- Before we start building Classification model, we should check how good is the size of the dataset we have
- Given balanced dataset, the next most important aspect of goodness is size of the data set
- The model should be able to converge during learning the parameters from the dataset
- If not, appropriate measure should be taken and care must be given while reporting performance
- We will see learning curve analysis that best suit to detect goodness of the size of the training dataset

Tips: Dataset Size

What to do with small data?

- Having small data but balanced can be approached in different ways to relay on the performance
- Note that the total data set we have will be divided into three for training, testing and validation
- The following are the techniques to minimize the effect of the dataset size

1. **k-fold cross validation:** randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a test set, and the method is fit on the remaining $k - 1$ folds.
2. **Data augmentation**: techniques used to increase the amount of data by adding slightly modified copies of already existing data

Tips:Dataset Size

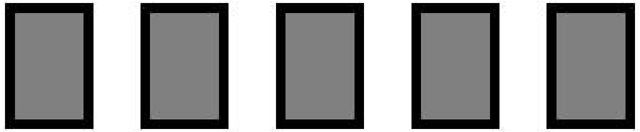
What to do with small data: Using K-fold cross validation--10-fold is the recommended

example:

– Break up data into groups of the same size

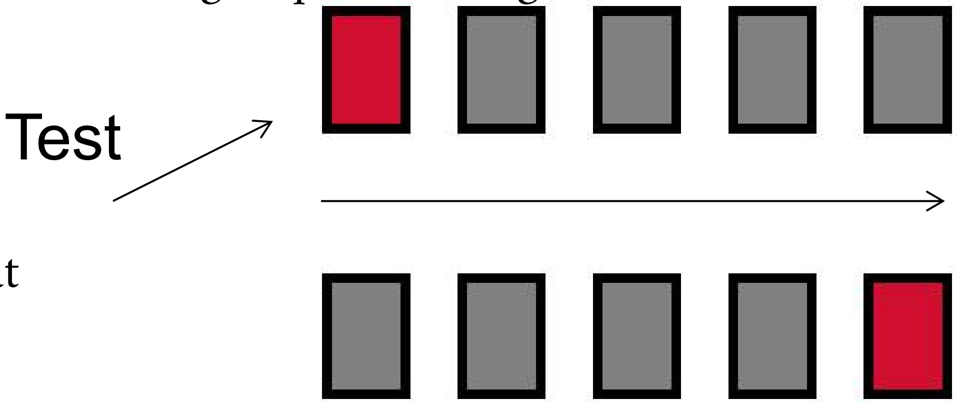
–

–



– Hold aside one group for testing and use the rest to build model

–



– Repeat

Feature Selection

- Why we need *Feature Selection (FS)?*
  - to improve performance (in terms of speed, predictive power, simplicity of the model).
  - to visualize the data for model selection.
  - To reduce dimensionality and remove noise.

- Feature Selection is a process that chooses an optimal subset of features according to a certain criterion.
- Given a set of **n** features, the goal of feature selection is to select a subset of **k** features (**k** < **n**) in order to minimize the classification error.

Feature Selection

- FS can be considered as a search problem.
- Search Directions (the two common):
  - **Sequential Forward selection(SFS):** In SFS variant features are sequentially added to an empty set of features until the addition of extra features does not reduce the criterion.
  - Mathematically if the input data in the algorithm is

    Input: $Y = \{y_1, y_2, \ldots, y_d\}$

  - Then the output will be :

    Output: $X_k = \{x_j \mid j = 1, 2, \ldots, k;\ x_j \in Y\}$, where $k = (0, 1, 2, \ldots, d)$

  - Where the selected features are k and K<d.
  - In the initialization X is a null set and k=0 (where k is the size of the subset).
  - In the termination, the size is k = p where p is the number of desired features.

Feature Selection

- Search Directions (the two common):
  - **Sequential Backward Selection(SBS):** SBS picks all the features from the input data and combines them in a set and sequentially removes them from the set until the removal of further features increases the criterion.
  - mathematically if the input data is

    Input: $Y = \{y_1, y_2, \ldots, y_d\}$

  - The output of the variant will be

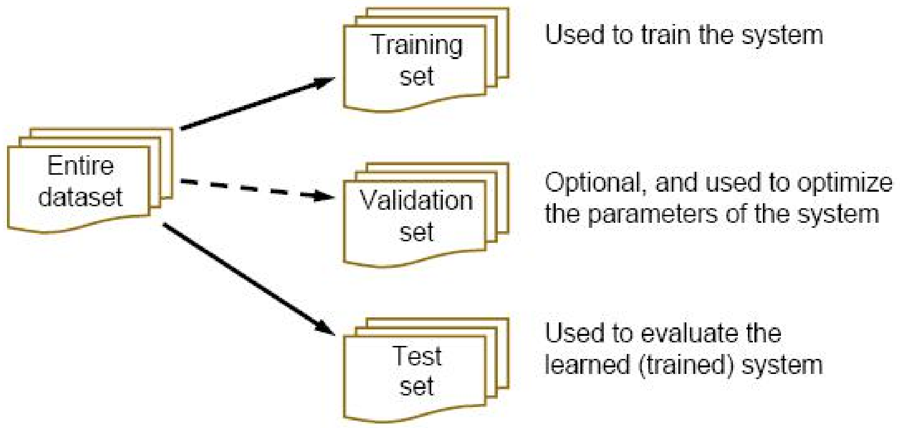
    Output: $X_k = \{x_j \mid j = 1, 2, \ldots, k;\ x_j \in Y\}$, where $k = (0, 1, 2, \ldots, d)$

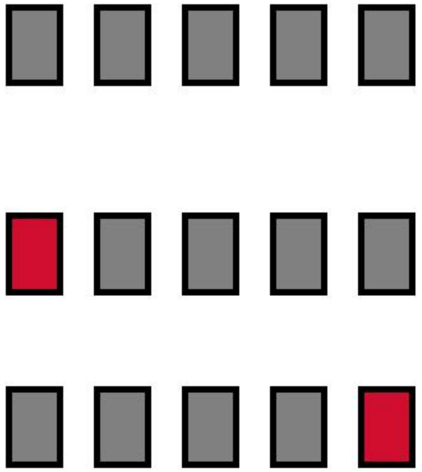
  - In the initialization X is a subset of features and k=d (where k is the size of the subset).
  - In the termination, the size is k = p where p is the number of desired features.

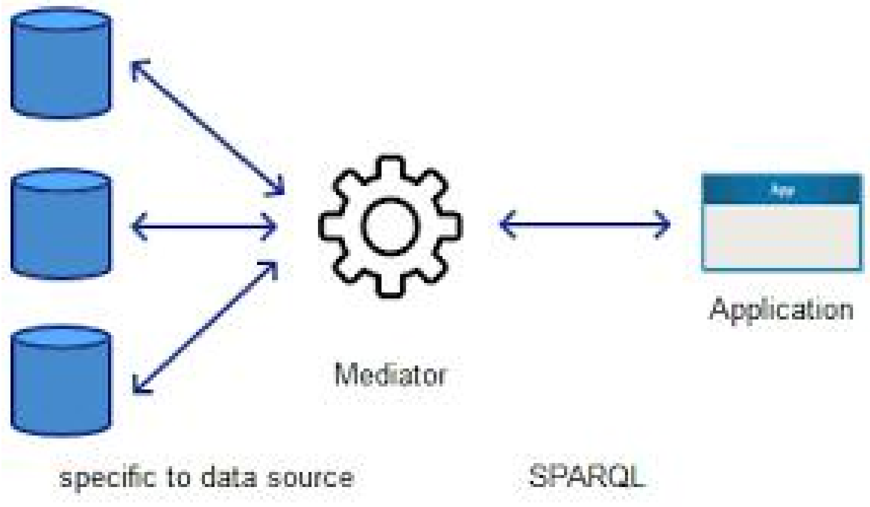
Feature Selection

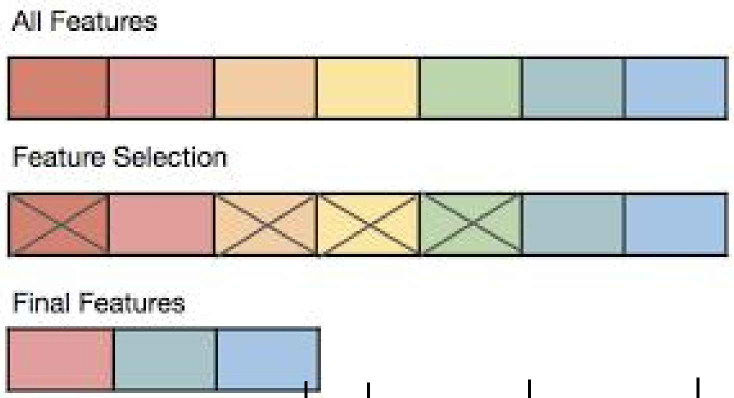
- Do you think that feature selection is different from dimensionality reduction?

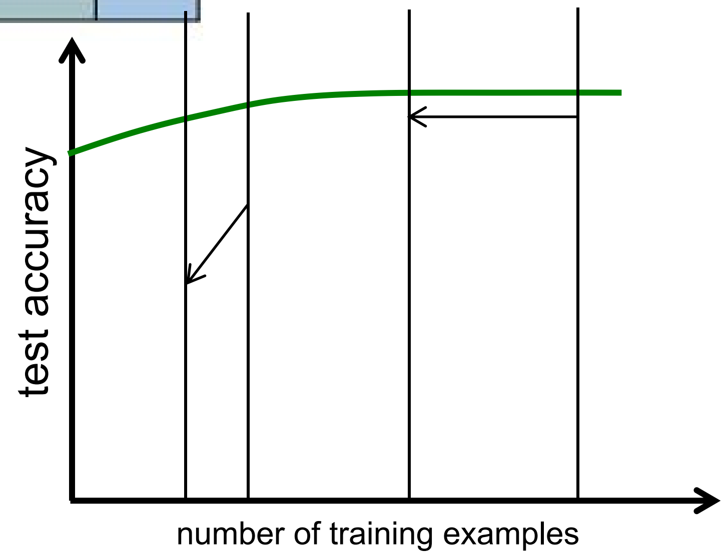
- Feature Selection:
  - When classifying novel patterns, only a small number of features need to be computed (i.e.,faster classification).
  - The measurement units (length, weight, etc.) of the features are **preserved**.
- Dimensionality Reduction:
  - When classifying novel patterns, all features need to be computed.
  - The measurement units (length, weight, etc.) of the features are **lost.**

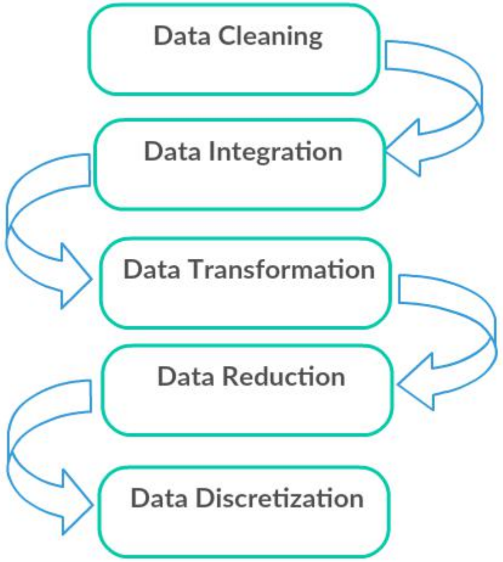

Entire dataset
Training set
Used to train the system
Validation set
Optional, and used to optimize the parameters of the system
Test set
Used to evaluate the learned (trained) system




Mediator
Application
specific to data source
SPARQL


All Features
Feature Selection
Final Features


test accuracy
number of training examples


Data Cleaning
Data Integration
Data Transformation
Data Reduction
Data Discretization