Data Analytics: Machine Learning

Topic 2

Machine Learning – Process & Metric

Assoc. Prof. DR UMI KALSOM YUSOF

SCHOOL OF COMPUTER SCIENCES

UNIVERSITI SAINS MALAYSIA (USM)

Methodologies for Machine Learning

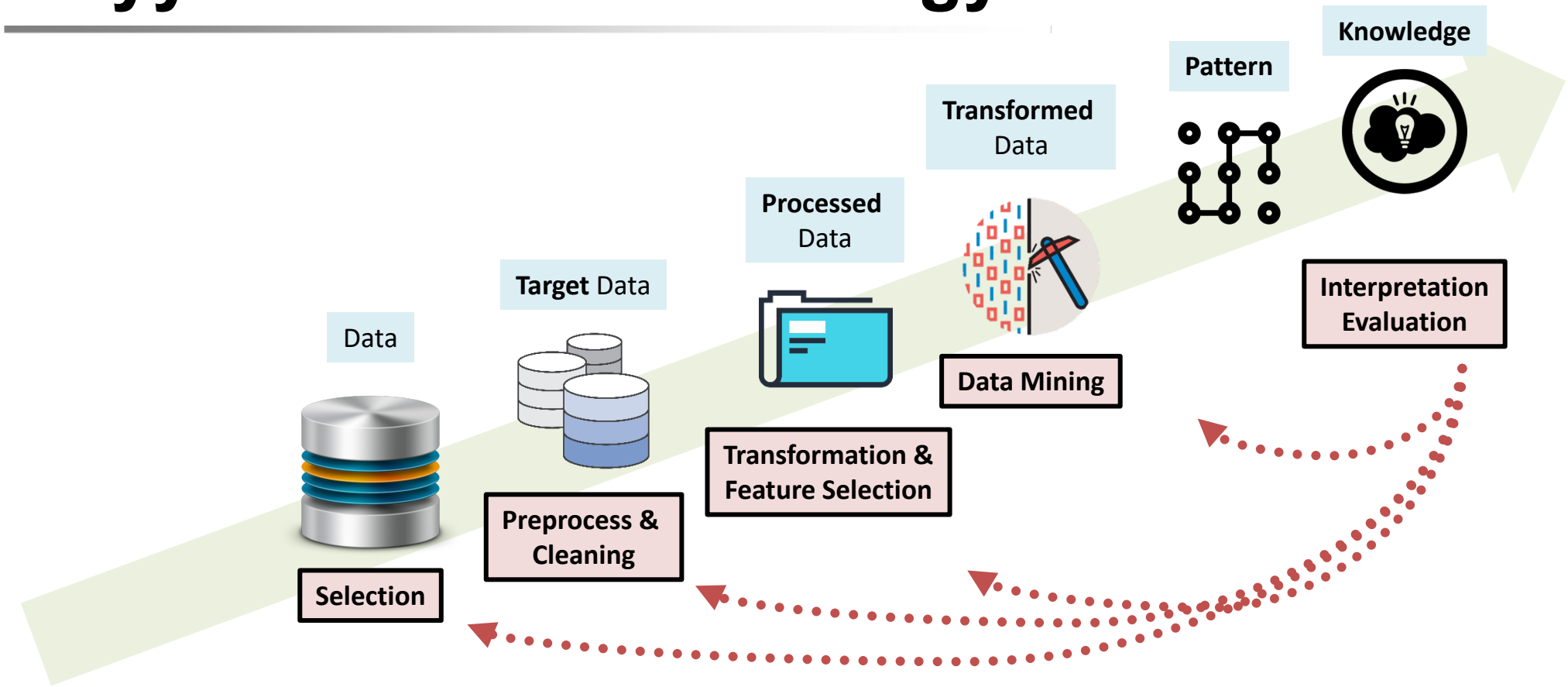
Fayyad's Knowledge Discovery in Databases (KDD)

Data Mining Methodologies

Sample, **E**xplore, **M**odify, **M**odel, **A**ssess (**SEMMA**)

CRoss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining (**CRISP-DM**)

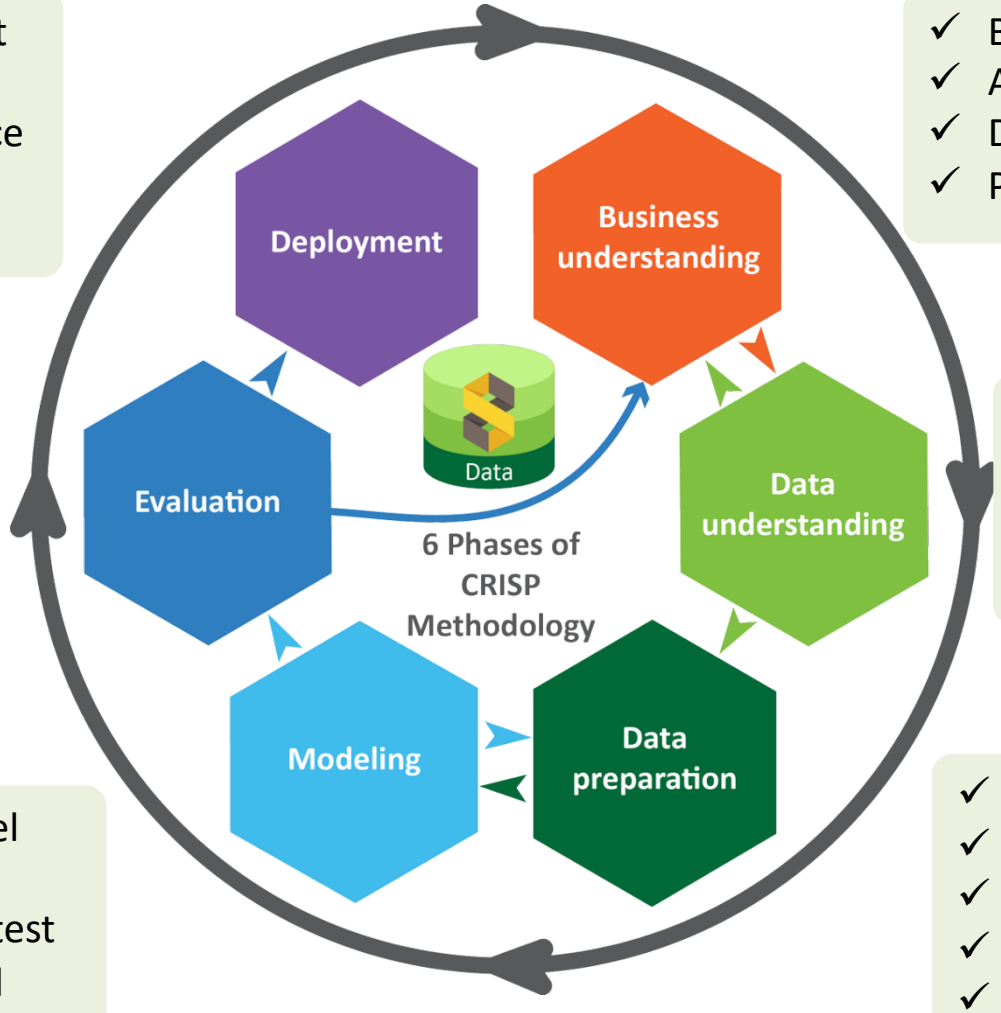
Fayyad's KDD Methodology



CRISP-DM Methodology

- ✓ **Plan** deployment
- ✓ **Plan** monitoring
- ✓ **Plan** maintenance
- ✓ **Final** report
- ✓ **Review** project

- ✓ Business **objective**
- ✓ Assess **situation**
- ✓ Data mining **goals**
- ✓ Project **plan**



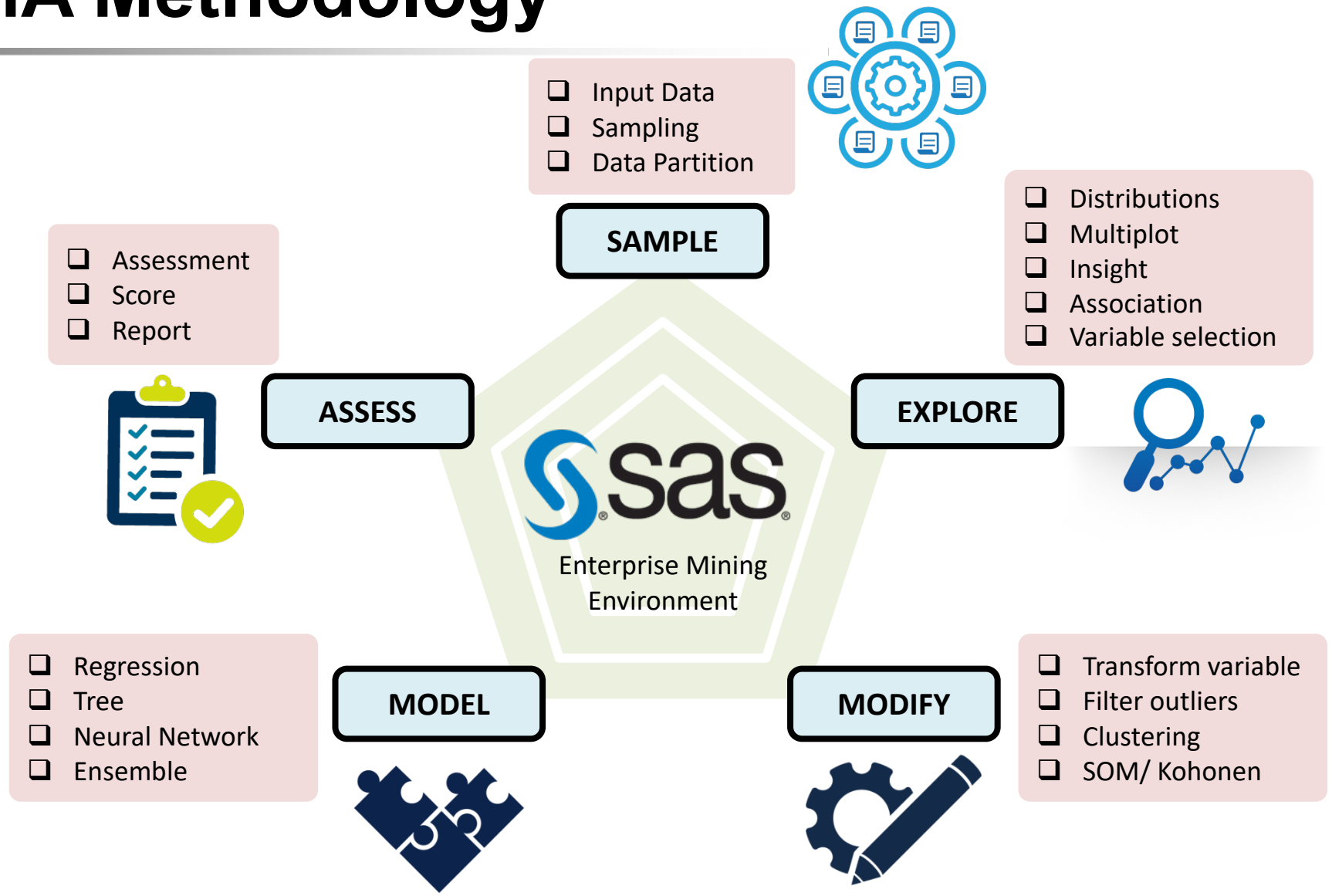
- ✓ **Evaluate** results
- ✓ **Review** process
- ✓ **Determine** next steps

- ✓ **Collect** data
- ✓ **Describe** data
- ✓ **Explore** data
- ✓ **Verify** data quality

- ✓ **Select** model
- ✓ **Techniques**
- ✓ **Design** the test
- ✓ **Build** model
- ✓ **Assess** model

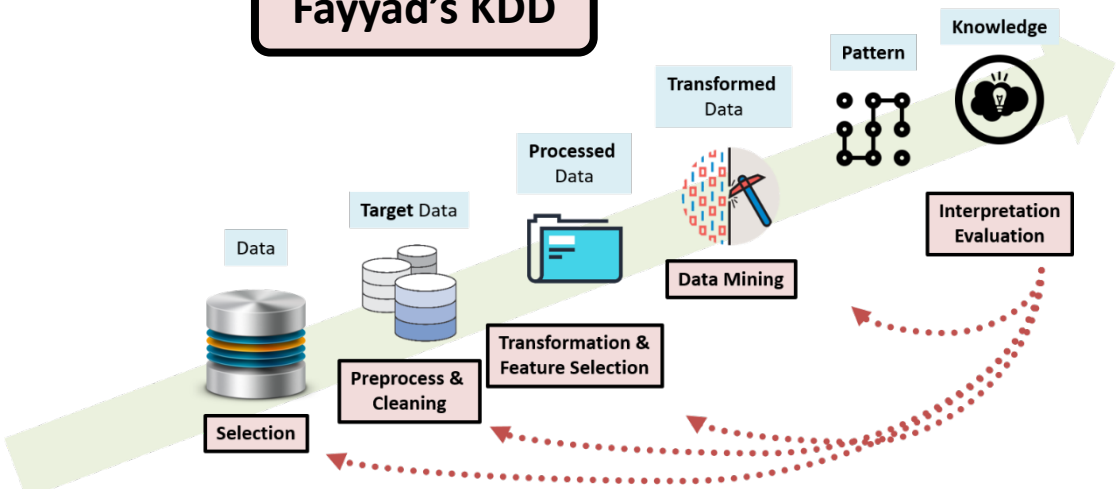
- ✓ **Select** data
- ✓ **Clean** data
- ✓ **Construct** data
- ✓ **Integrate** data
- ✓ **Format** data

SEMMA Methodology

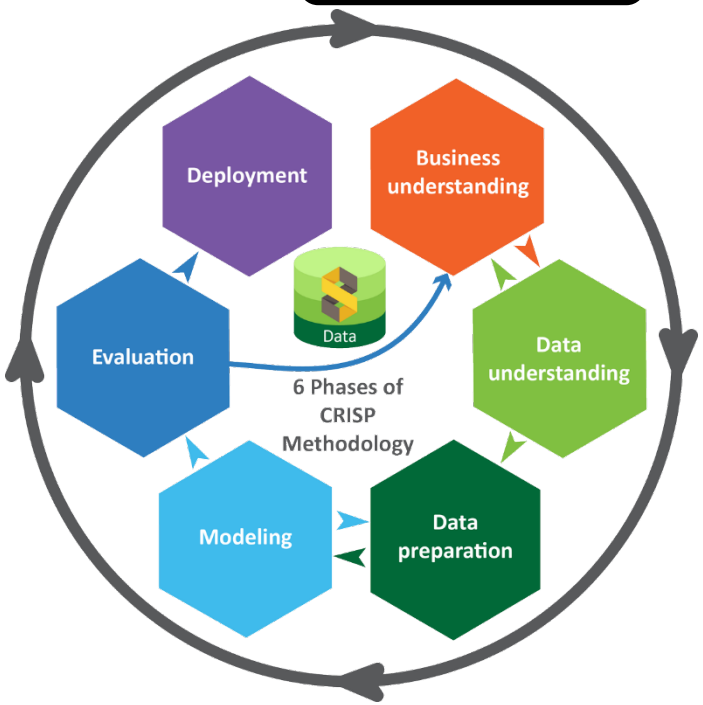


A Comparison

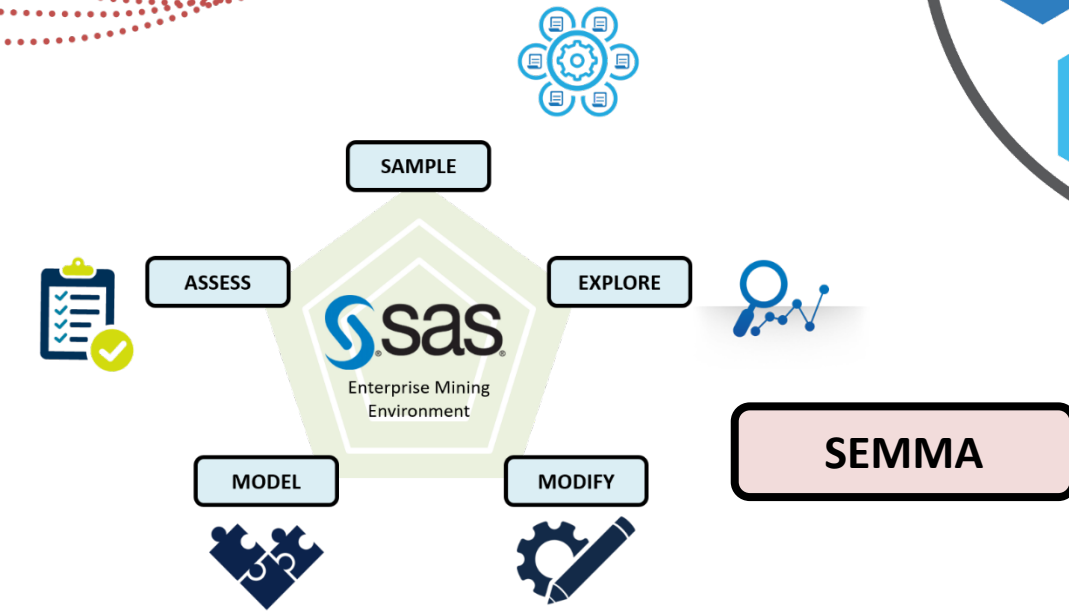
Fayyad's KDD



CRISP-DM



SEMMA



CRISP-DM Process

Why CRISP-DM

Data Mining Process	❑ **Reliable** & **Repeatable** ❑ **Little** Data Mining **Skills**
Uniform Framework	❑ Guidelines ❑ Experience Documentation
Account for Flexibility	❑ Different **business/agency** ❑ Different **data**

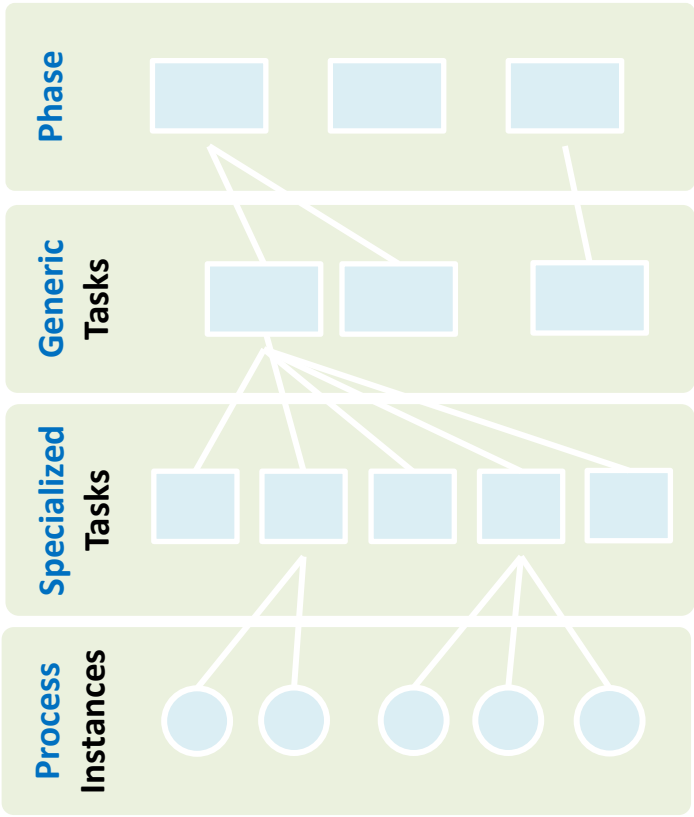
Features

- ❑ **Non-proprietary**
- ❑ **Application/Industrial** neutral
- ❑ **Tool** neutral
- ❑ Focus on **business issues**
- ❑ Focus on **technical analysis**
- ❑ **Framework** for **guidance**
- ❑ **Experience** base
- ❑ **Templates** for analysis

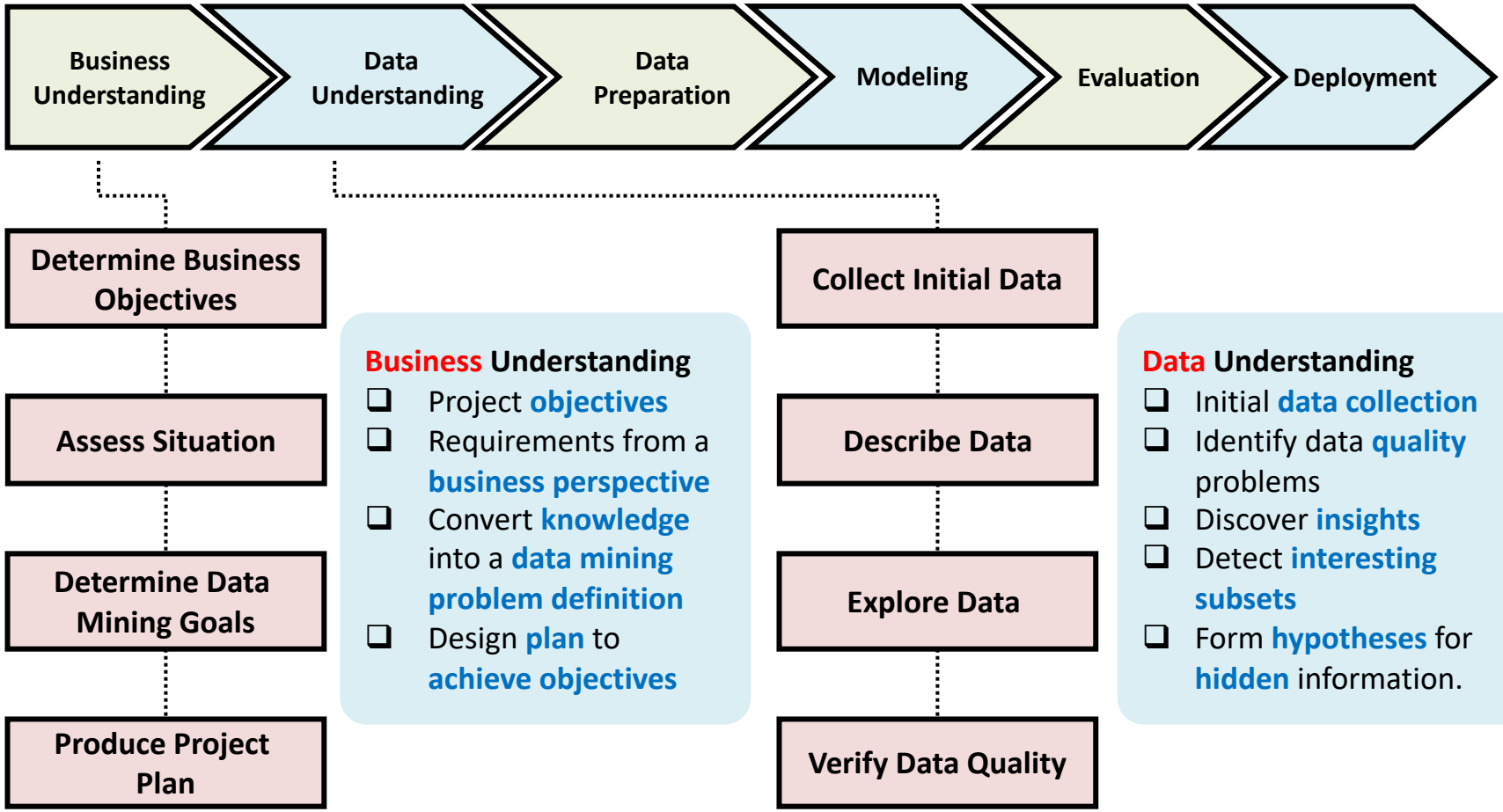
Hierarchical Process Model

Set of tasks

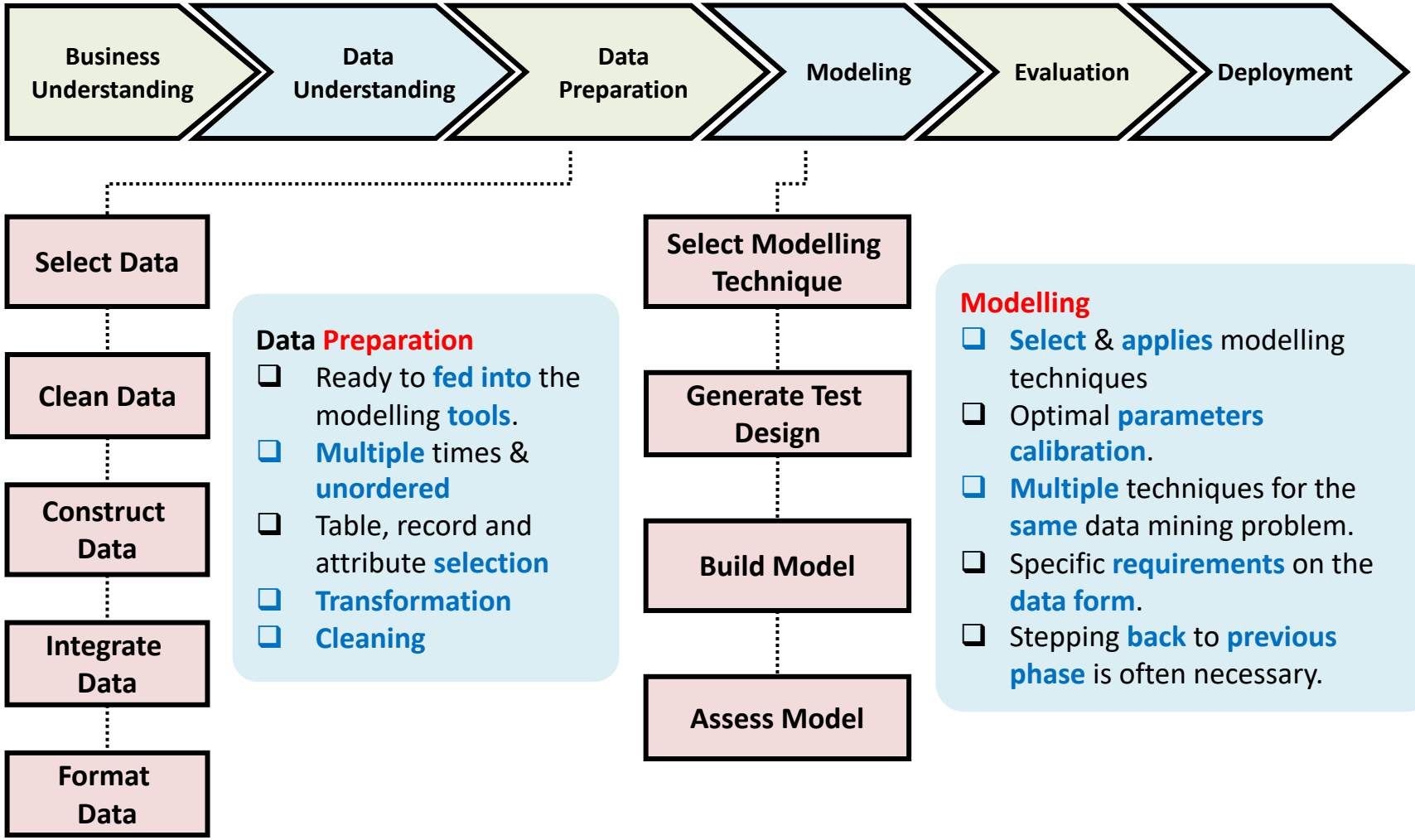
4 levels of abstraction



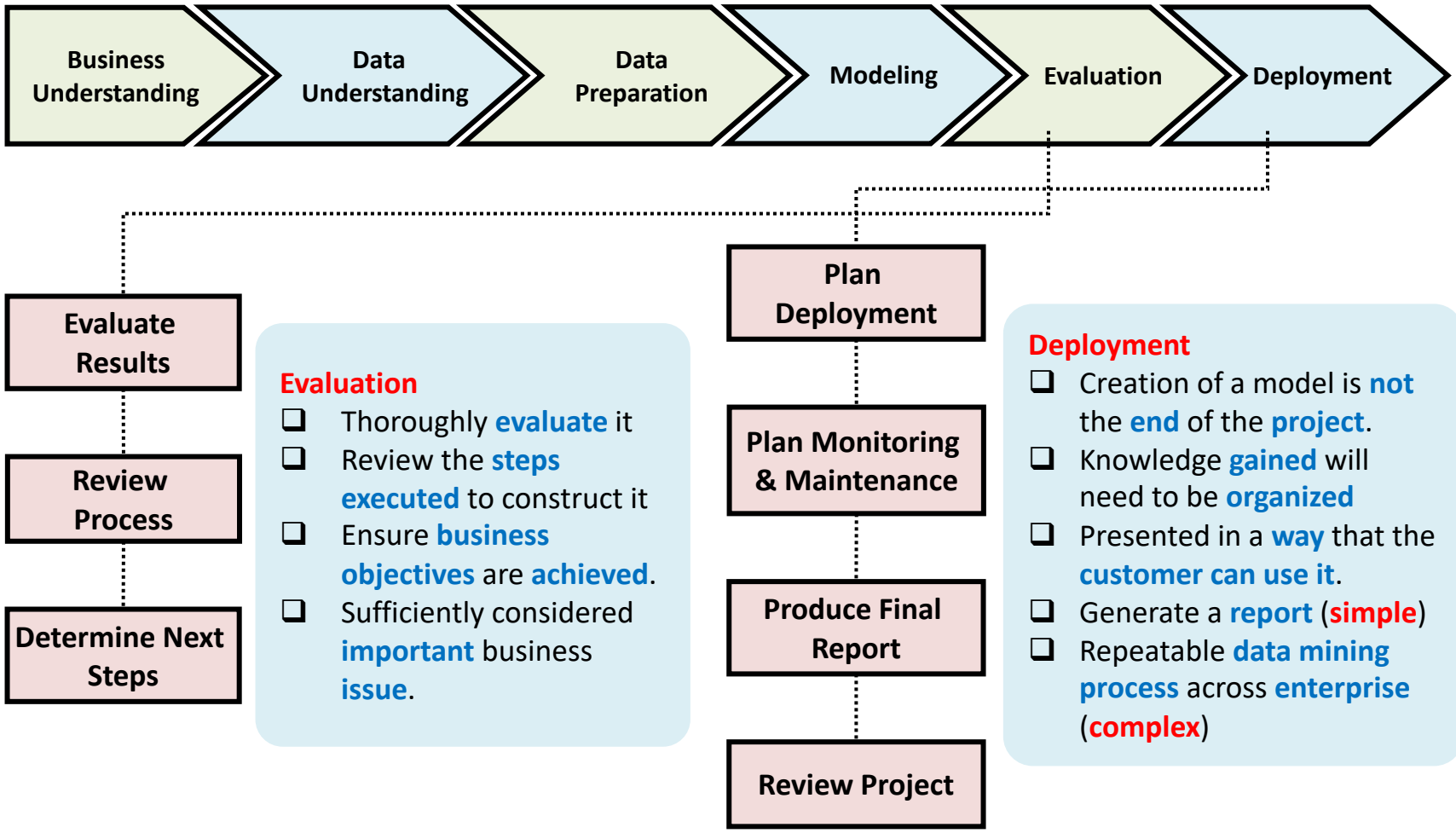
Phases & Generic Tasks



Phases & Generic Tasks



Phases & Generic Tasks



Motivation

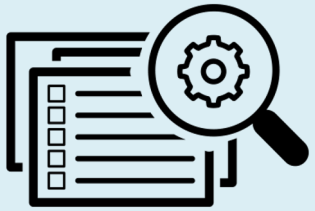
Important to evaluate classifier's generalization performance:

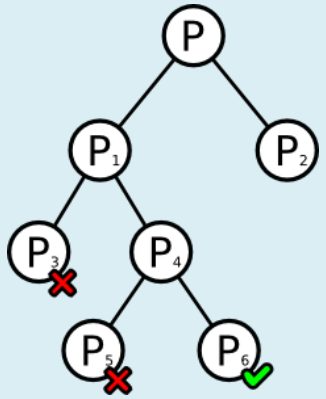
Determine whether to **employ** the **classifier**

Example:
Learning the **effectiveness** of medical treatments from a **limited-size data**, it is important to **estimate** the **accuracy** of the **classifiers**

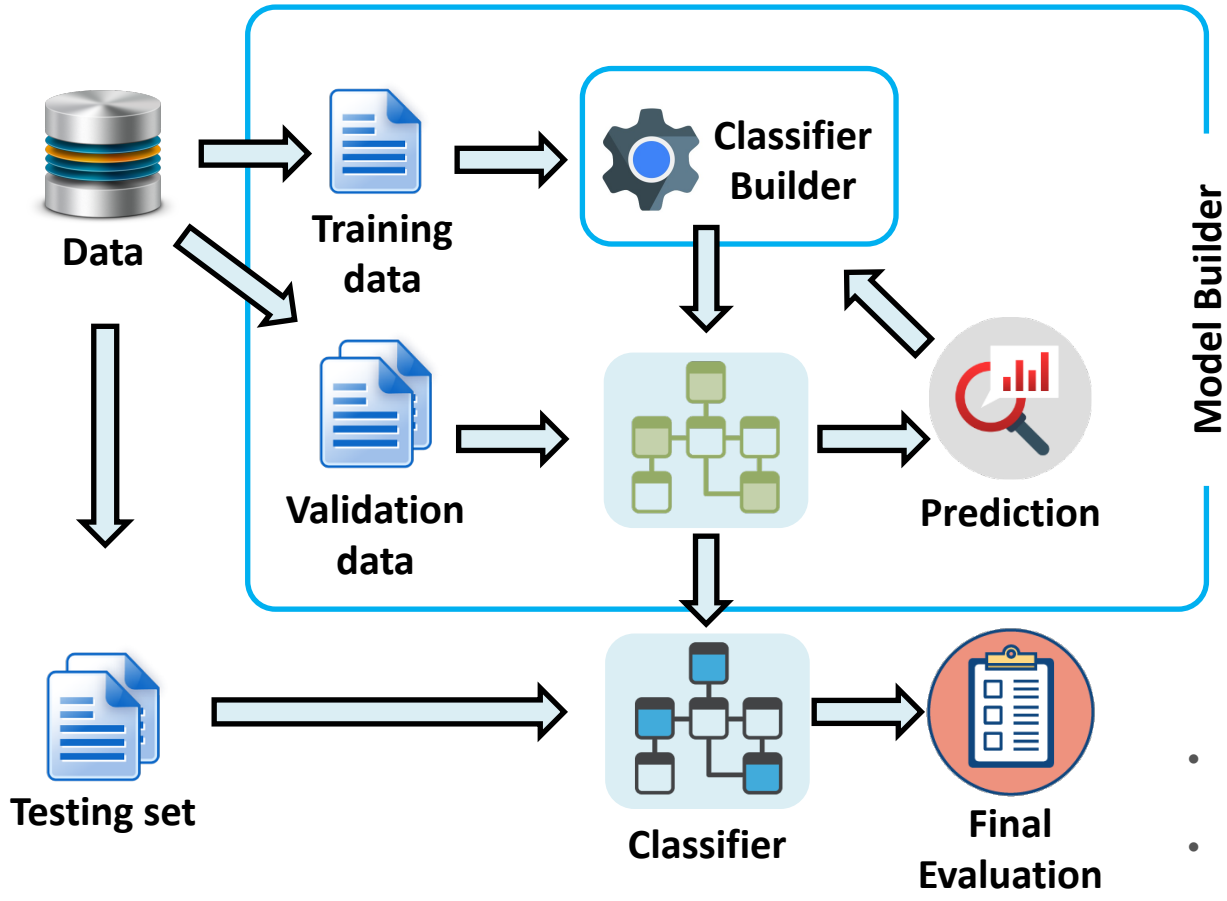
Optimize the **classifier**

Example:
when **post-pruning** decision trees we must **evaluate** the **accuracy** of the decision trees on each **pruning step**





Classification:
Train, Validation, Test Split



The test data can't be used for parameter tuning!

Making The Most of The Data

Once evaluation is **complete**, **all the data** can be used to **build** the **final classifier**.

The **larger** the **training data** the **better** the classifier (but returns diminish).

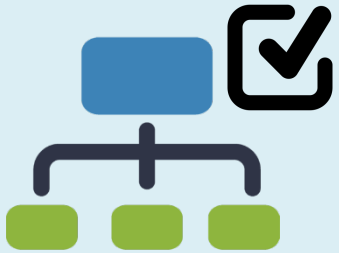
The **larger** the **test data** the more **accurate** the error estimate.

- ***Training set***: *A set of examples used for learning, that is to fit the parameters of the classifier.*
- ***Validation set***: *A set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network.*
- ***Test set:*** *A set of examples used only to assess the performance of a fully-specified classifier.*

How to Estimate the Metrics?

To estimate the metrics (accuracy), we can use below as (model evaluation techniques):

- **Training** data;
- **Independent** test data;
- **Hold-out** method;
- **k-fold** cross-validation method;
- **Leave-one-out** method;
- **Bootstrap** method;
- and many more…



*Model Evaluation techniques – To increase the performance of the model pertaining dataset

Data Mining
Metrics Evaluation

Training Data

The accuracy/metric estimates on the training data are not good indicators of performance on future data

New data will probably not exactly the same as the training data!

This measure the degree of classifier's overfitting (or underfitting).

Independent Test Data

- Used when we have plenty of data
- Natural way of forming training and test data

Example:

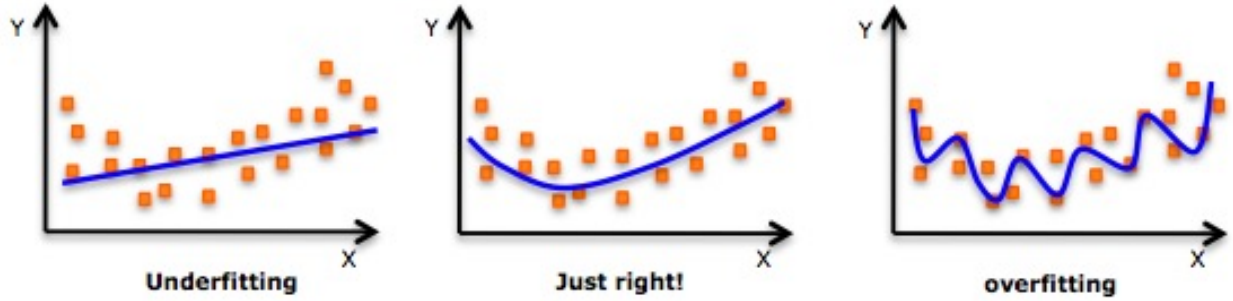
- Quinlan in 1987 reported experiments in a medical domain
- Trained on data from 1985
- Tested on data from 1986.

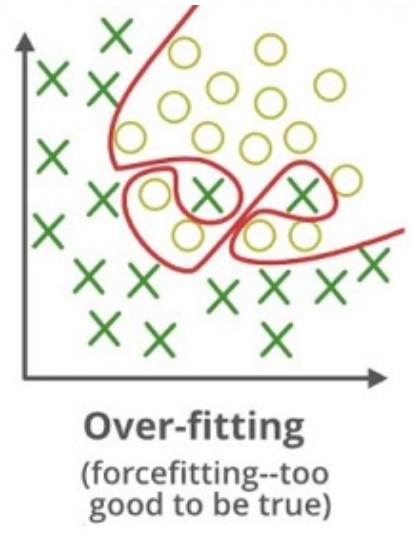
What is Overfitting?

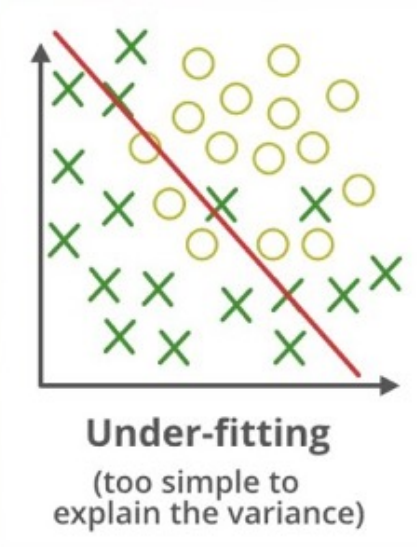
Overfitting & Underfitting

Overfitting: A model that models the training data too well.

Underfitting: A model that can neither model the training data nor generalize to new data

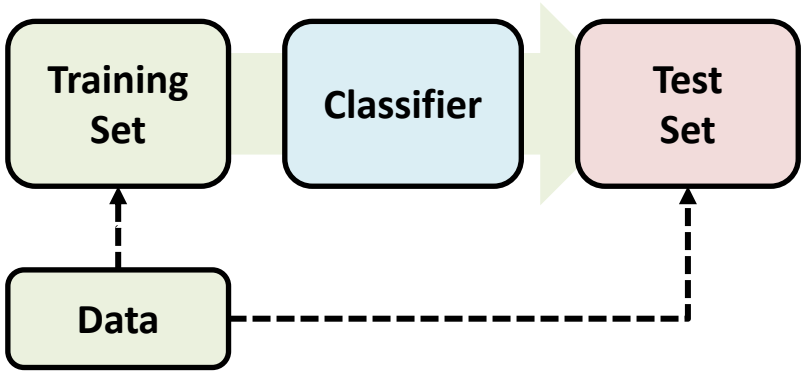






Hold-out Method

Hold-out

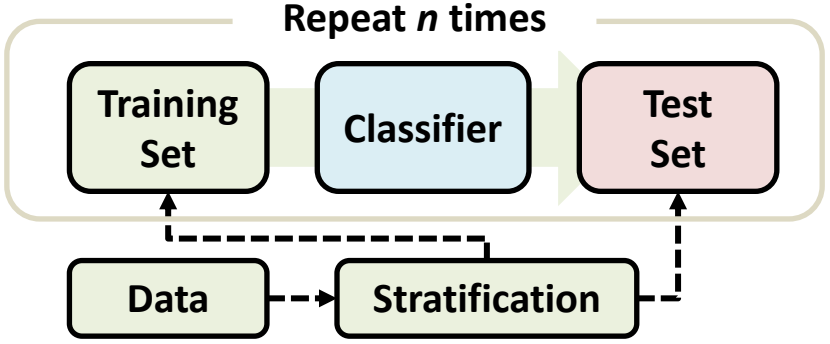


The method **splits** the data into **training** data and **test** data (usually **2/3 for train**, **1/3 for test**).

Then, **build** a classifier using the train data and **test** it using the test data.

Used when there are **thousands** of **instances** & several **hundred instances** from each **class**.

Repeated Hold-out



Holdout estimate can be made **more reliable** by **repeating** the process with **different subsamples**.

In each iteration, a certain **proportion** is **randomly** selected for **training** (possibly with **stratification**).

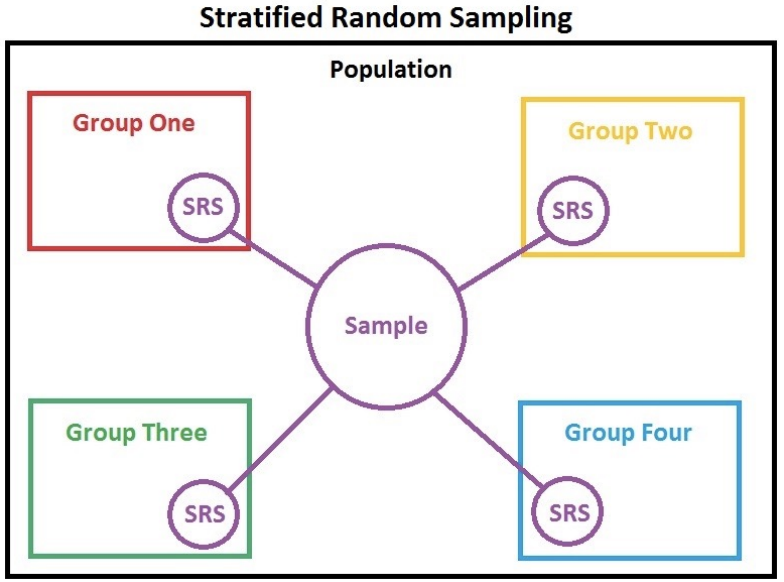
The **error rates** on the **different** iterations are **averaged** to **yield** an **overall error rate**.

Stratification

The **holdout** method **reserves** a certain amount for **testing** and uses the **remainder** for **training**.

For "**unbalanced**" datasets, samples might not be representative (Few instances or none of some classes).

Stratified sample
An advanced version of balancing the data where each class is represented with approximately equal proportions



Stratification is the process of **dividing members** of the **population** into **homogeneous subgroups** before **sampling**.

The strata should be mutually exclusive

Every **element** in the population **must** be assigned to **only one stratum**.

The strata should be collectively exhaustive

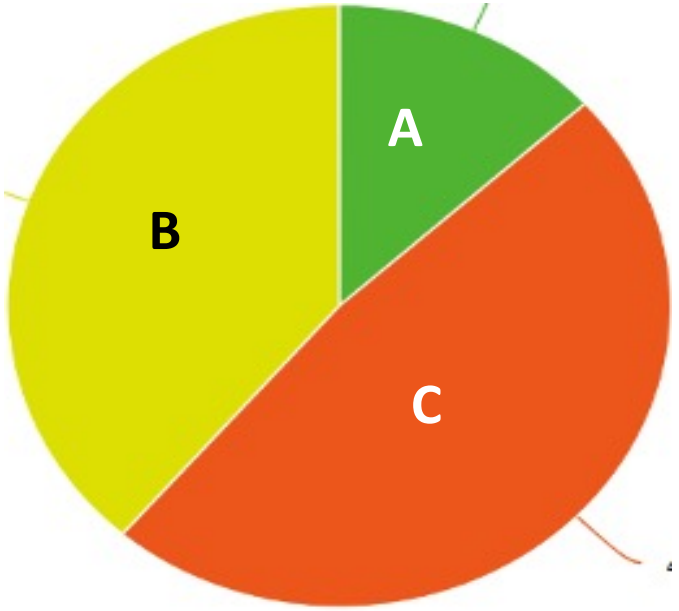
- No population **element** can be **excluded**.
- **Simple random sampling** is applied or;
- **Systematic sampling** is applied

This often **improves** the **representativeness** of the sample by **reducing sampling error**.

Stratification

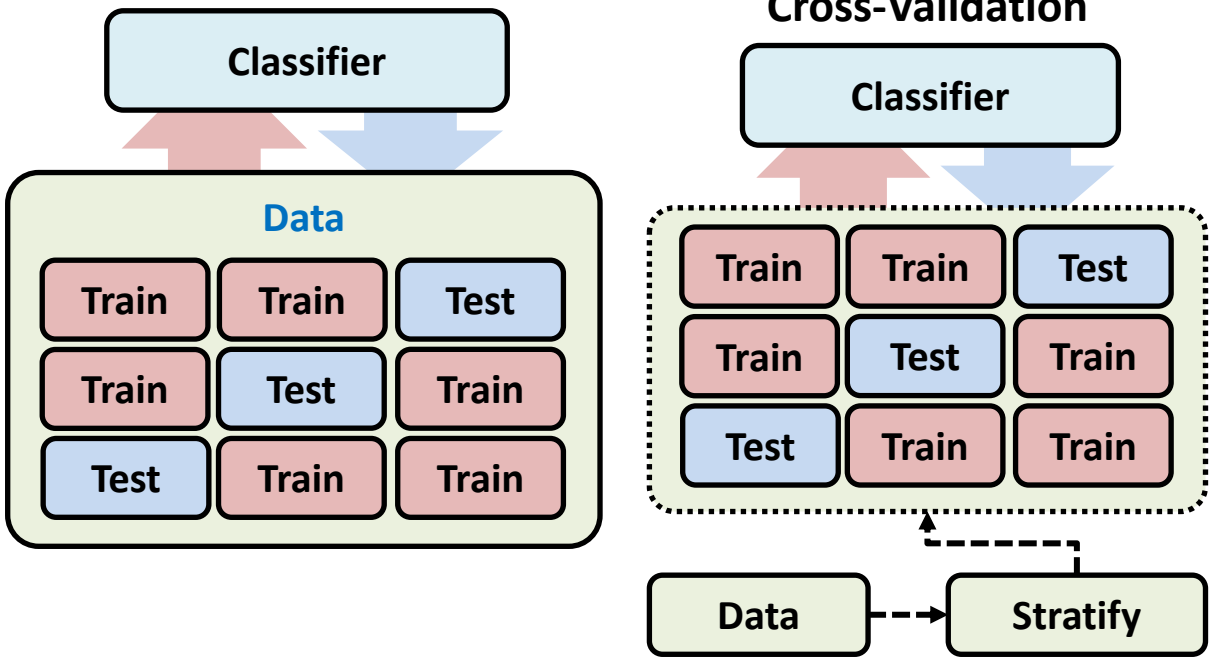
Example

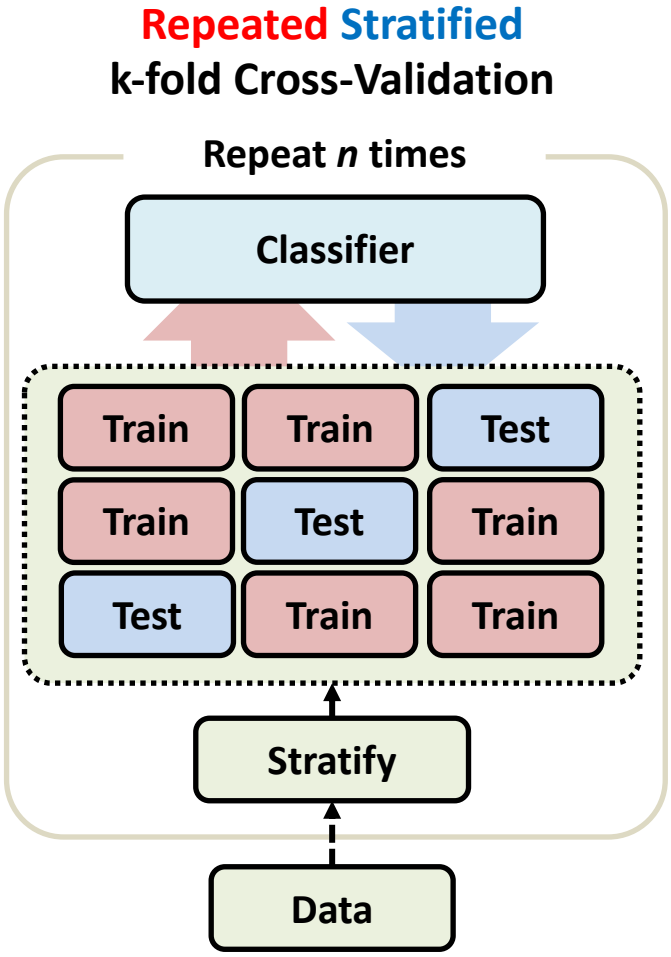
- Need to estimate average number of votes for each candidate in an election. The country has 3 towns:
  - Town A has 1 million factory workers
  - Town B has 2 million office workers
  - Town C has 3 million retirees.



- A **random sample of size 60** over entire population will have some chance of:
  - The random sample not well balanced across these towns
  - Bias causing a significant error in estimation.
- **Random sample** of 10, 20 and 30 from Town A, B and C respectively
- Produce a smaller error in estimation for the same total size of sample.

k-Fold Cross-Validation





- Avoids **overlapping test sets** data is **equally split** into ***k* subsets**
- Some subset for **testing**, remainder for **training**.
- Recommended **k = 10** (**best choice**)
- Results are **averaged** (**reduces** the **estimate's variance**)
- **Standard**: **Stratified** *k*-fold cross-validation.
- **Even better**: **Repeated stratified** *k*-fold cross-validation.

Leave-One-Out Cross-Validation

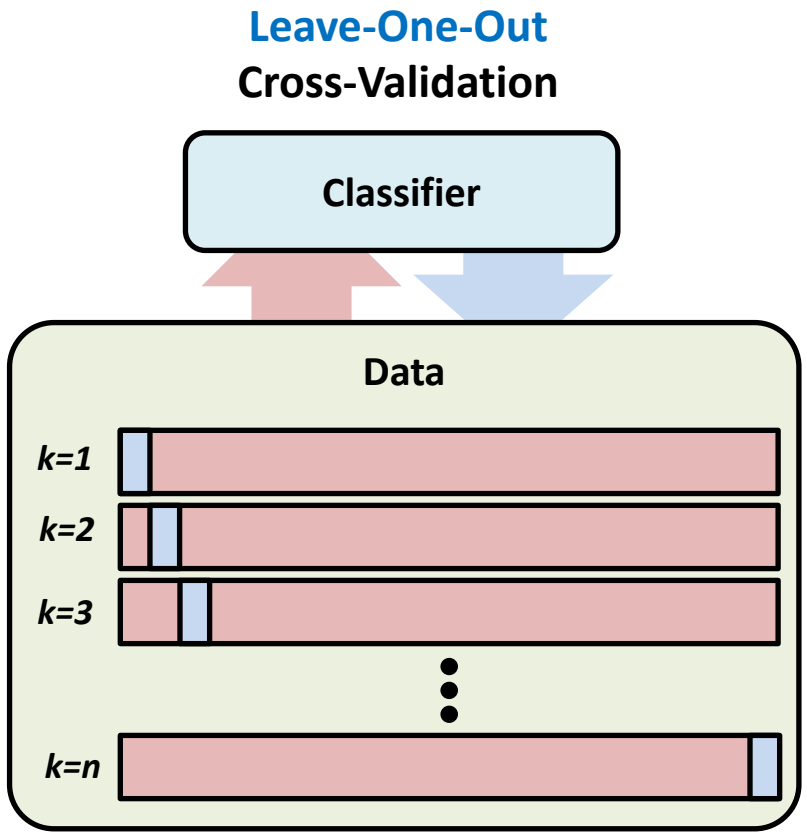
An extreme form of cross-validation

- Set number of folds to number of training instances;
- N-1 training instances, 1 test instance, and build classifier N times.

Make best use of the data

Involves no random sub-sampling

Very computationally expensive



Bootstrap Method

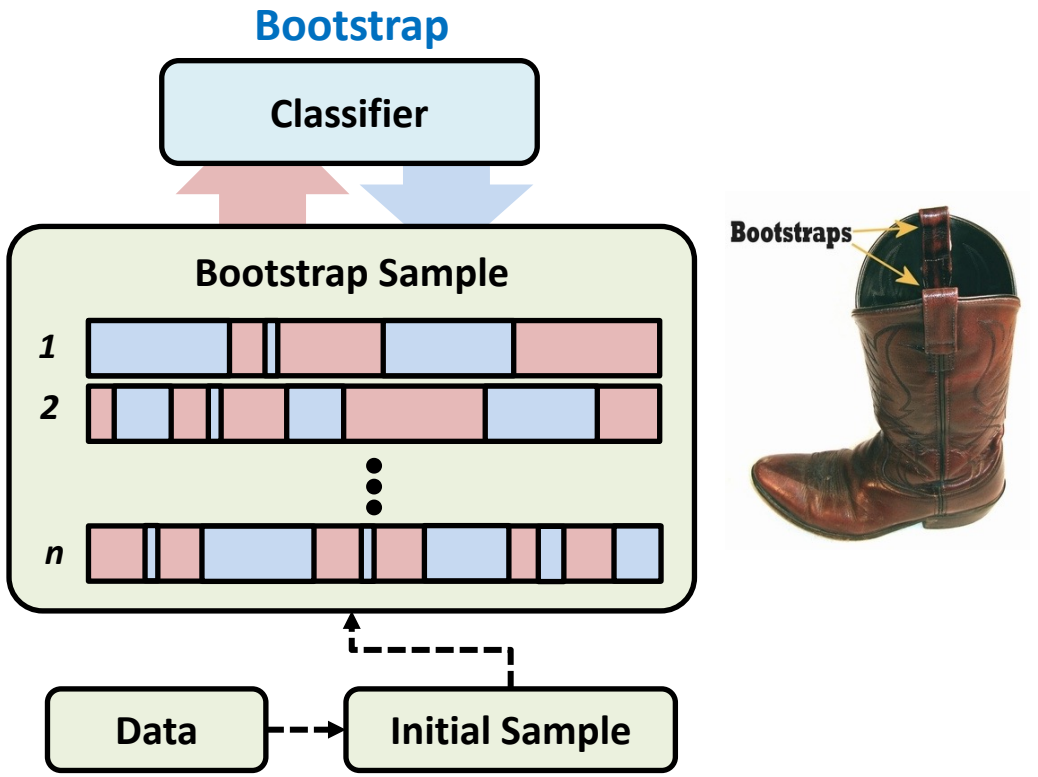
- Cross-validation uses sampling without replacement
- The same instance, once selected, cannot be selected again for a particular training/test set

Bootstrap Method

- Uses sampling with replacement to form the training set
- Sample a dataset of *n* instances *n* times with replacement to form a new dataset of *n* instances
- Use this data as the training set
- Use the instances from the original dataset that don't occur in the new training set for testing.

A bootstrap sample is a smaller sample that is "bootstrapped" from a larger sample.

- Bootstrapping is a type of resampling
- Large numbers of smaller samples of the same size are repeatedly drawn, with replacement, from a single original sample.



Bootstrap Method

Sampling with Replacement

- You want to find the probability of some event where you replace the item each time you choose one.
- Let's say you had a population of 5 people, and you wanted to sample
- Their names are:
  - John
  - Jack
  - Qiu
  - Tina
  - Hatty

1. Put their names in a hat.
2. If you **sample with replacement**, choose one person's name
3. Put that person's name back in the hat
4. Then choose another name.

The **possibilities** of **two-name sample** are:

- John, John
- John, Jack
- John, Qui
- Jack, Qui
- Jack Tina
- …and so on.

Metric Evaluation

Summary

- Use test sets and the hold-out method for “large” data;
- Use the cross-validation method for “middle-sized” data;
- Use the leave-one-out and bootstrap methods for small data;
- Don’t use test data for parameter tuning - use separate validation data.

Classification measures

- Accuracy is only one measure (error = 1 - accuracy).
- Accuracy is not suitable in some applications.
- In text mining, we may only be interested in the documents of a particular topic, which are only a small portion of a big document collection.
- In classification involving skewed or highly imbalanced data, e.g., **network intrusion** and **financial fraud detections**, we are interested only in the minority class.
  - High accuracy does not mean any intrusion is detected.
  - E.g., 1% intrusion. Achieve 99% accuracy by doing nothing.
- The class of interest is commonly called the **positive class**, and the rest **negative classes.**

How to evaluate the Classifier's Generalization Performance?

- We test a classifier on some test set
- We derive at the end the following *confusion matrix*:

Actual class \ *Predicted class*	Pos	Neg	
Pos	*TP*	*FN*	*P*
Neg	*FP*	*TN*	*N*

Classification Measures

- Accuracy = $(TP+TN)/(P+N)$
- Error = $(FP+FN)/(P+N)$
- Precision = $TP/(TP+FP)$
- Recall/TP rate (Sensitivity) = TP/P
- Specificity = TN/N
- FP Rate = FP/N

precision can be understood as the probability that a randomly chosen predicted positive instance would be relevant

recall is how close we are to a specific target on average.

Predicted class

Actual class	Pos	Neg	Recall
Pos	*TP*	*FN*	*P*
Neg	*FP*	*TN*	*N*
Precision			

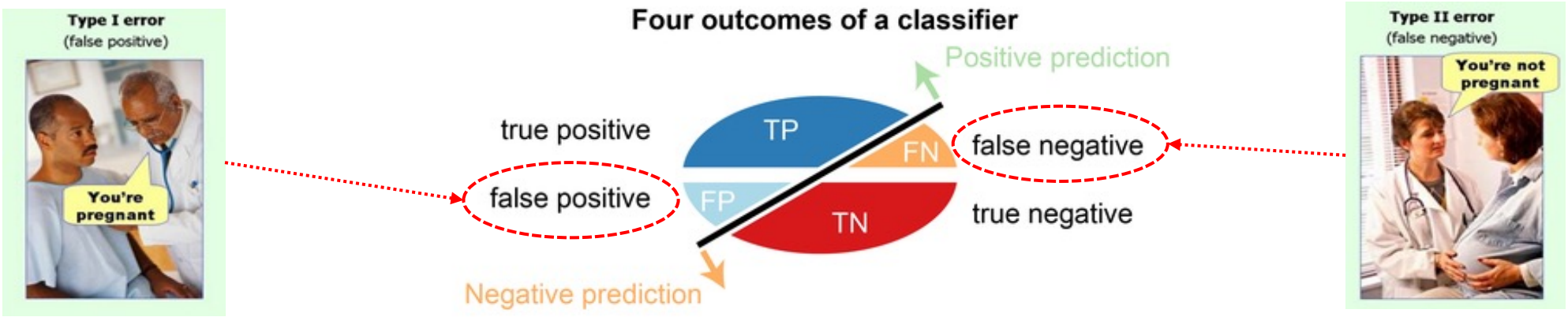
Example:

Predicted class

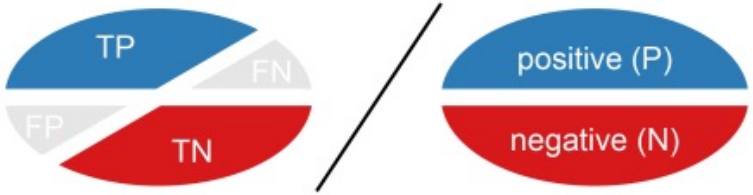
Actual class	Pos	Neg	
Pos	*100*	*20*	*120*
Neg	*30*	*50*	*80*
Precision	**130**		

- Accuracy = (100 + 50)/(120 + 80) = 150/200 = 75
- Error = (30 +20)/(120 + 80) = 50/200 = 25%
- Precision = TP/(TP+FP) = 100 / 130 = 76.9%
- Recall/TP rate = TP/P = 100/120 = 83.3%
- Specificity = TN/N = 50/80 = 62.5%
- FP Rate = FP/N = 30/80 = 37.5%

Classification Measures



Accuracy: (TP + TN) / (P + N)



Accuracy is calculated as the total number of two correct predictions (TP + TN) divided by the total number of a dataset (P + N).

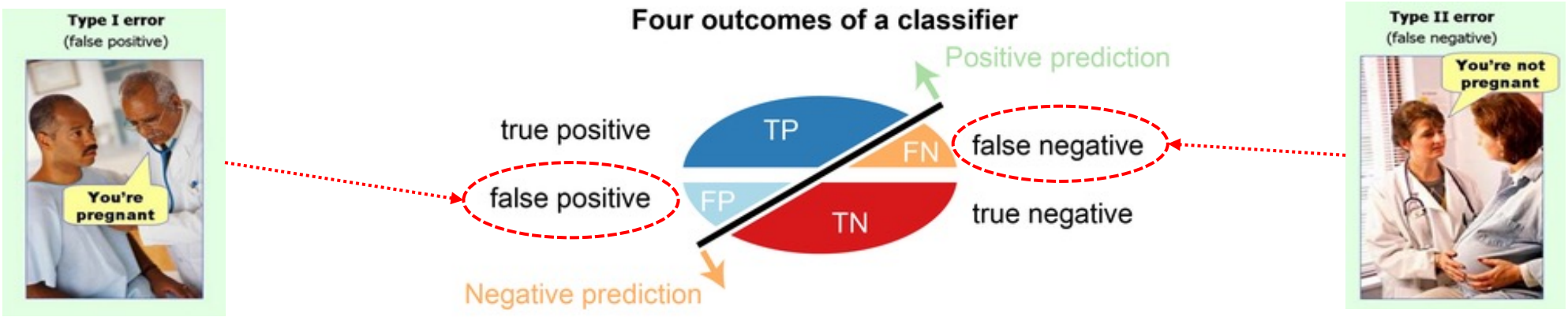
Error rate: (FP + FN) / (P + N)

Error rate is calculated as the total number of two incorrect predictions (FN + FP) divided by the total number of a dataset (P + N).

Precision: TP / (TP + FP)

Precision is calculated as the number of correct positive predictions (TP) divided by the total number of positive predictions (TP + FP).

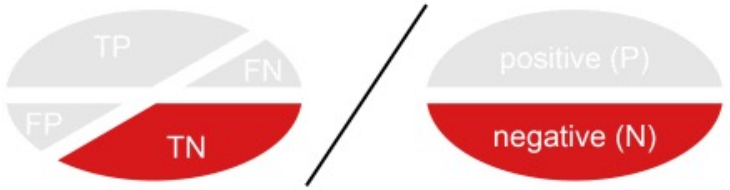
Classification Measures



Sensitivity: TP / P Recall

Sensitivity is calculated as the number of correct positive predictions (TP) divided by the total number of positives (P).

Specificity: TN / N



Specificity is calculated as the number of correct negative predictions (TN) divided by the total number of negatives (N).

False positive rate: FP / N

False positive rate is calculated as the number of incorrect positive predictions (FP) divided by the total number of negatives (N).

F_1-value (also called F_1-score)

- It is hard to compare two classifiers using two measures. F_1 score combines precision and recall into one measure

$$F_1 = \frac{2pr}{p+r}$$

F_1-score is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{p}+\frac{1}{r}}$$

- The harmonic mean of two numbers tends to be closer to the smaller of the two.
- For F_1-value to be large, both *p* and *r* much be large.

Cost Matrices & Classification

Goal of machine learning is based around making a computer **generalize** its **observation**.

The measure of performance of a machine learning algorithm is based on its accuracy of classifying a data set.

In practice, different types of classification errors often incur different costs

Examples:

- Terrorist **profiling**
  "Not a terrorist" correct 99.99% of the time
- Loan **decisions**
- Fault **diagnosis**
- **Promotional** mailing

Cost Matrices

True class \ Predicted class	Positive	Negative
Positive	TP Cost	FN Cost
Negative	FP Cost	TN Cost

*Usually, TP Cost and TN Cost are set equal to 0!

If the classifier outputs class probability, adjust it to minimize the expected prediction cost.

Expected cost is computed as dot product of class probabilities vector with appropriate column in cost matrix.

Cost Matrices & Classification

Example

Assume that a classifier returns for an instance probs $p_{pos} = 0.6$ and $p_{neg} = 0.4$.

The expected cost if the instance is classified as positive:
0.6 * 0 + 0.4 * 10 = 4

The expected cost if the instance is classified as negative:
0.6 * 5 + 0.4 * 0 = 3

Simple methods for cost sensitive learning

- **Resampling** of instances according to **costs**;
- **Weighting** of instances according to **costs**.

True class \ Predicted class	Positive	Negative
Positive	0	5
Negative	10	0

True class \ Predicted class	Positive	Negative
Positive	0.6 * 0	0.6 * 5
Negative	0.4 * 10	0.4 * 0

ROC Curves

Stands for "receiver operating characteristic"

Used in signal detection to show tradeoff between hit rate and false alarm rate over noisy channel

ROC curves are similar to lift charts

Differences to lift chart

- y axis shows percentage of true positives in sample rather than absolute number
- x axis shows percentage of false positives in sample rather than sample size

- It is commonly called the ROC curve.
- It is a plot of the true positive rate (TPR) against the false positive rate (FPR).
- True positive rate:

$$TPR = \frac{TP}{TP + FN}$$

- False positive rate:

$$FPR = \frac{FP}{TN + FP}$$

ROC Curves and Analysis

- A ROC (Receiver Operating Characteristics) curve :

Classifier 1

True	Predicted	
	pos	neg
pos	40	60
neg	30	70

Classifier 2

True	Predicted	
	pos	neg
pos	70	30
neg	50	50

Classifier 3

True	Predicted	
	pos	neg
pos	60	40
neg	20	80

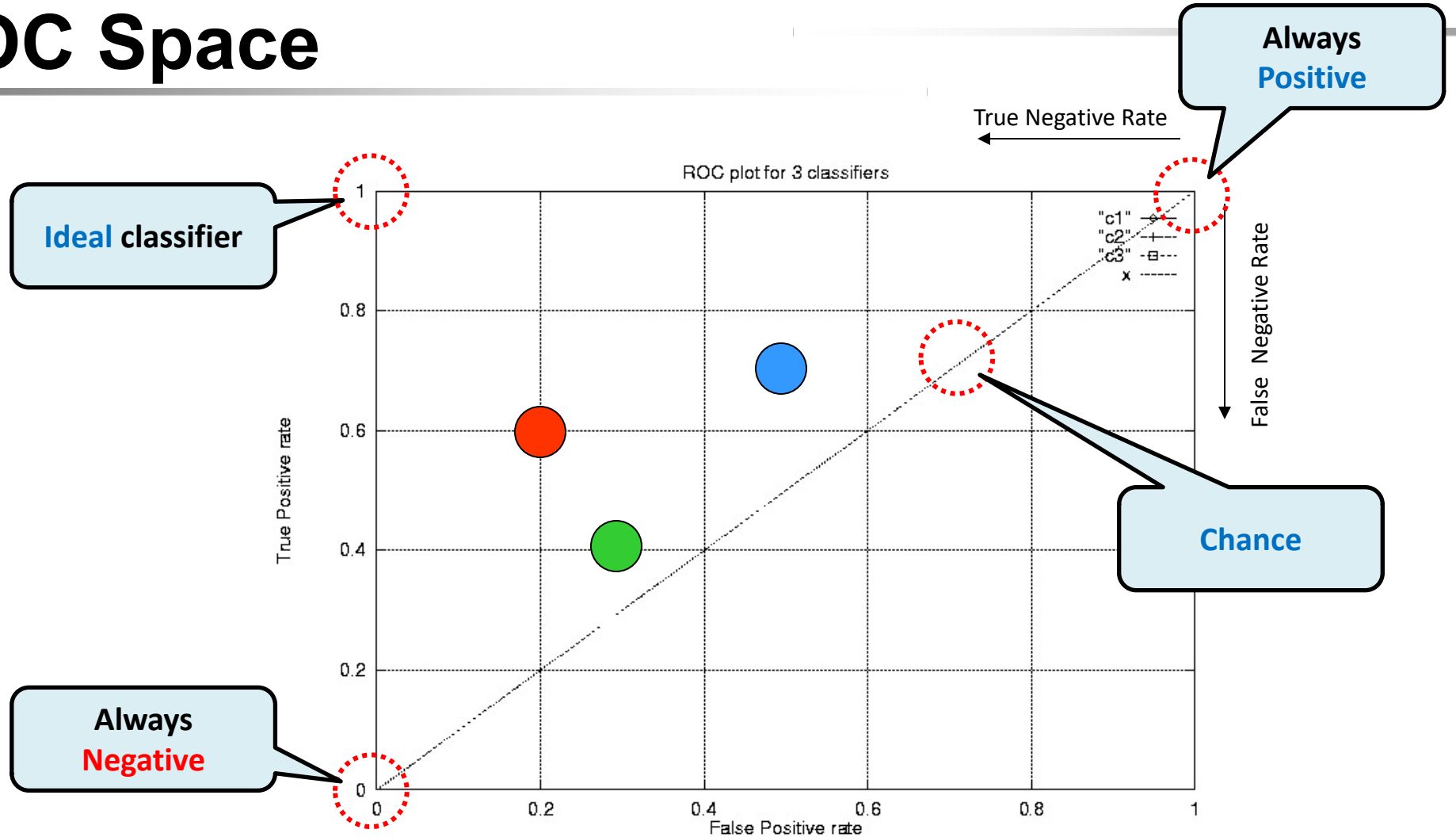
Classifier 1
TPr = 0.4
FPr = 0.3

Classifier 2
TPr = 0.7
FPr = 0.5

Classifier 3
TPr = 0.6
FPr = 0.2

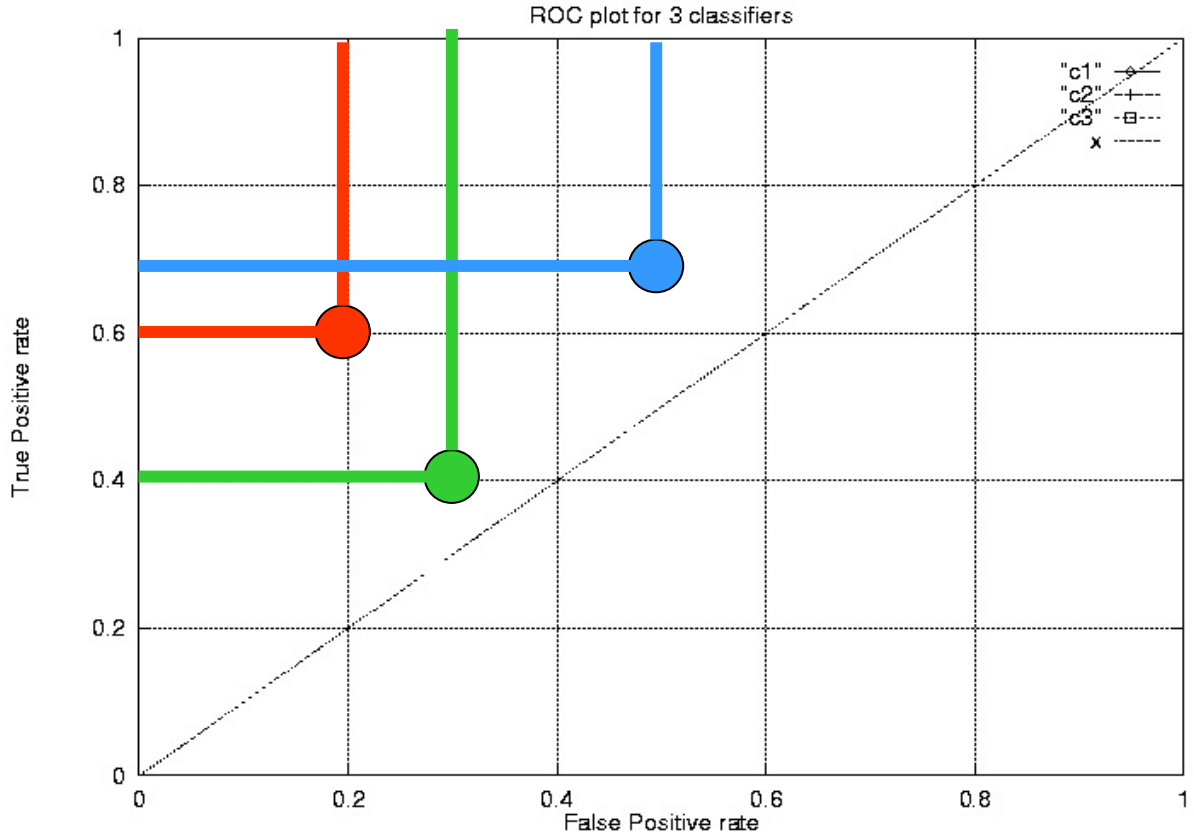
TPr – True Positive rate
FPr – False Positive rate

ROC Space


Always
Positive
True Negative Rate
ROC plot for 3 classifiers
Ideal classifier
"c1"
"c2"
"c3"
x
False Negative Rate
True Positive rate
Chance
Always
Negative
False Positive rate
0
0.2
0.4
0.6
0.8
1

Dominance in the ROC Space

Which one is the best option?



Classifier A dominates classifier B if and only if $TPr_A > TPr_B$ *and* $FPr_A < FPr_B$.

Thank you