CUNY School of Professional Studies

SPS.CUNY.EDU

2020 Fall Data-622
Introduction to Machine Learning and Big Data
Raman Kannan

Instructor Email Address: Raman.Kannan@sps.cuny.edu

Acknowledgements:
Generous support from IBM Power Systems Academic Initiative
IBM PSAI provides computing infrastructure for free

Subject Matter of the Course

- Machine Learning: Birds Eye View
- Supervised Learning
  - KYD – Know your dataset
    - IV/DV -- Multivariate
    - Quantitative/Qualitative
    - Scaling/Normalization
    - Categorical:Encoding
    - Class Labels - Imbalances
    - Training/Testing
    - Over/Underfitting
    - Bias/Variance
  - Process Discipline
    - Repeatability/reproducibility
    - NFL and Occams Razor
- Parametric Classifiers
  - Logistic
  - Naïve Bayes
  - Discriminat Analysis
- Non Parametric Classifiers
  - Instance Based :kNN
  - Rule Based:Decision Tree
  - Geometric (topological):SVM
- Analytical Foundation of Classifiers
  - MLE
  - Optimization:Loss Function
  - Convergence: GD, SGD
- Comparative Analysis of Classifiers
  - Sources of Error, Estimation
    - Bias vs Variance
    - Error Reduction Strategies
      - Resampling Methods
      - Penalization Methods
      - Cross Validation
- Ensemble Methods
  - Bagging/Boosting
  - Combining Like/Unlike
- Learners
  - ML, Big Data, Parallelism

How will this class run?

100% Online
Weekly One hour Call
Weekly Reading Assignment/Contribute Discussion Forum
Instructor Led Clinical session
- 90% of the work in R programming language
  students may do their work in python if they wish
- 10% shell scripting (bash) and remote computing

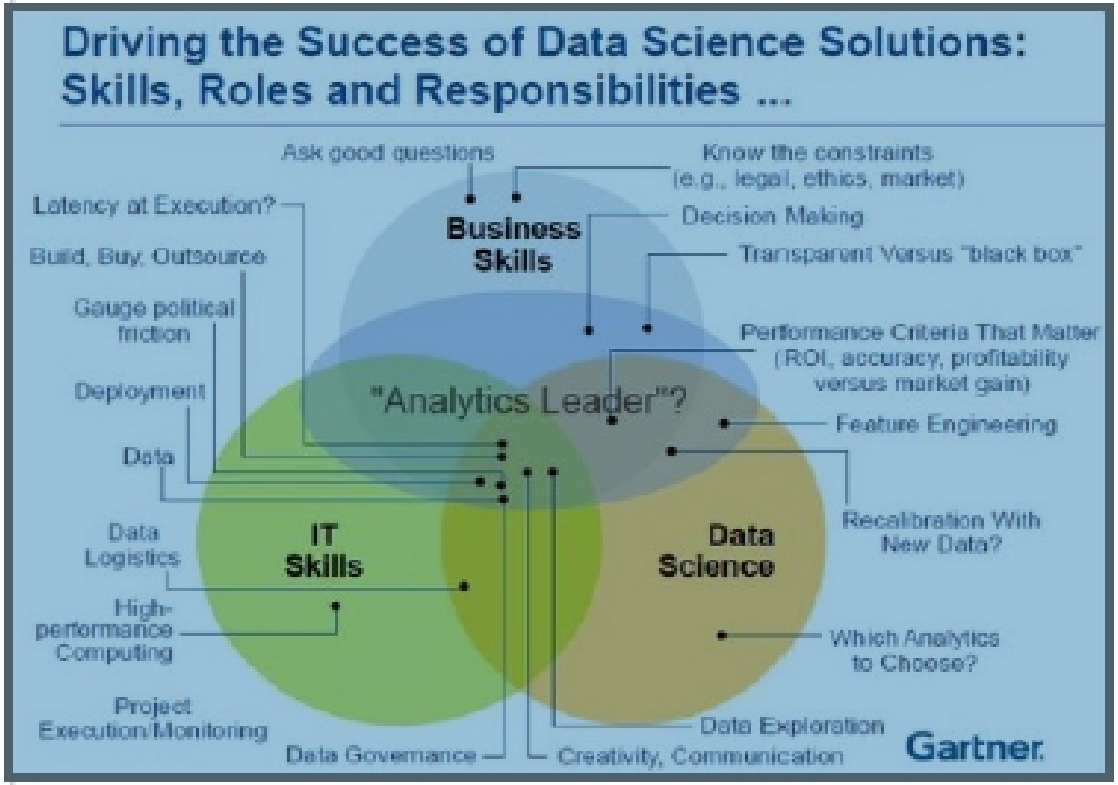
Assessments	Grading Scheme
Read the syllabus	
2 individual modeling assignments	30%
Engaging and Participation	12%
Combination of weekly knowledge evaluation, quiz, ML Tasks	40%
Open book/open notes test – t1	9%
Open book/open notes test – t2	9%
follow process discipline&on time	Late submission not graded

Machine Learning

- Machine: automation/objective/endurance
  - patently human, we seek to automate
- Learning: Observe/infer/improve/feedback
  - Learning is how we evolve/improve in life.

M/L is a sub-discipline of AI, confluence of Computing & Statistics
M/L shares numerous topics with data mining, pattern recognition
M/L, therefore, shares topics from Data Science and Statistical Learning
M/L employs data science techniques, linear algebra,calculus and probability and statistics.
To be masterful in M/L one has to be a competent/efficient software engineer –able to gather,prepare, process, analyze data and most importantly convey the findings to a broader audience.

What is M/L ?



Boundary is amorphous.
It is part of AI, always has been.
Many other related areas such as Data Mining/Data Science have evolved that today the distinction is further blurred.

M/L has co-opted Neural Nets and NLP.

But there is no debate on the need to integrate Business, technology and Data Science to achieve M/L.

Difference between Machine Learning, Data Science, AI, Deep Learning, and Statistics – Vincent Granville
https://www.datasciencecentral.com/profiles/blogs/difference-between-machine-learning-data-science-ai-deep-learning
https://mlplatform.nl/what-is-machine-learning/
https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/
https://en.wikipedia.org/wiki/Timeline_of_machine_learning

Historical Perspective

> Over the past 50 years the study of Machine Learning has grown from the efforts of a handful of computer engineers exploring whether computers could learn to play games, and a field of Statistics that largely ignored computational considerations, to a broad discipline that has produced fundamental statistical-computational theories of learning processes, has designed learning algorithms that are routinely used in commercial systems for speech recognition, computer vision, and a variety of other tasks, and has spun off an industry in data mining to discover hidden regularities in the growing volumes of online data.
>
> – Tom Mitchell, CMU Machine Learning Department

To be more precise, we say that a machine learns with respect to a particular task T, performance metric P, and type of experience E, if the system reliably improves its performance P at task T, following experience E. Depending on how we specify T, P, and E, the learning task might also be called by names such as data mining, autonomous discovery, database updating, programming by example, etc. – also credited to Tom Mitchell

"The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience."
-- Tom Mitchell

https://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf

Definition of M/L

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.-- *https://en.wikipedia.org/wiki/Machine_learning*

A scientific field is best defined by the central question it studies. The field of Machine Learning seeks to answer the question
“How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?”

M/L Tasks

Supervised learning,
Unsupervised learning,
Semi-supervised learning
Reinforcement learning

Our focus however is

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.[1] It infers a function from *labeled training data* consisting of a set of *training examples*.[2] In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way (see inductive bias).

The parallel task in human and animal psychology is often referred to as concept learning.

Course focus

Our objective develop vocabulary around critical concepts central to practicing Supervised Learning, aka Classifiers or classification algorithms.

We will develop required expertise to:

- prepare dataset,
  - Load
  - Clean (missing values, range, outlier, unbalanced)
  - Exploratory Analysis (discriptive statistics)
  - Feature selection/scale/transform
  - run one or more classifiers
    - training phase
    - test/validation phase
- write a report on summary findings.

https://www.r-bloggers.com/the-real-prerequisite-for-machine-learning-isnt-math-its-data-analysis/

Classifiers In Scope

Lessons from Linear Regression

Parametric Classifier:

- Logistic Regression
- Naive Bayes
- Fisher's Discriminant Analysis LDA (when variances are same)
- Quadratic Discriminat Analysis QDA (when variances are different)

Non-Parametric

- ***Instance based*** – kNN –Nearest Neighbor
- ***Logic Based*** – Decision Tree
- ***Geometry (topology) based*** – Support Vector Machine

Combining Classifiers

- Bagging
- Boosting
- Random Forest
- Stacking

We will prescribe and adopt a process and repeat the process
Using each of these algorithms – over the same dataset in R
heart.csv and ecoli.csv
We will adopt how we may measure the performance
Compare each of the above classifiers using that measure

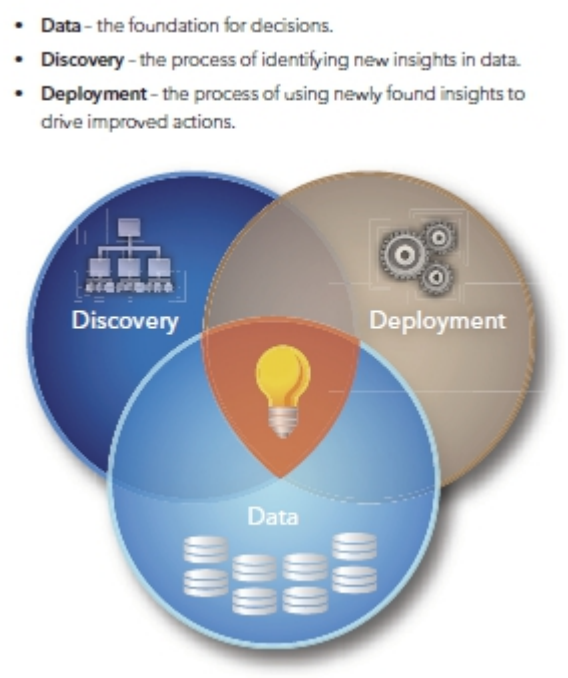
https://www.datasciencecentral.com/profiles/blogs/top-10-machine-learning-algorithms

Dataets heart.csv and ecoli.csv are provided on blackboard

Reading on Introduction

https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf
https://amueller.github.io/COMS4995-s19/slides/aml-01-introduction/#p43
https://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf
https://www.r-bloggers.com/the-real-prerequisite-for-machine-learning-isnt-math-its-data-analysis/
https://www.datasciencecentral.com/profiles/blogs/top-10-machine-learning-algorithms
https://ciml.org
https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/data-mining-from-a-z-104937.pdf

Process



Data – the foundation for decisions.
• Discovery – the process of identifying new insights in data.
• Deployment – the process of using newly found insights to drive improved actions

https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/data-mining-from-a-z-104937.pdf

Know your data!

- Data
  - Set of observations, one record
  - Each observation is a set of
    - Attribute,fields,variables
    - One or more independent variable (IV)
    - One dependent variable (DV)

If the dependent variable(DV) is numerical/continuous → Regression
If the DV is categorical/nominal → Supervised Learning

When all the data relating to observation is in one record, dataset is said to be in wide format.The algorithms we will consider require wide format.

Structure of data

- Wide format – each row is an observation
- One or more dimensions (attributes or features)
- Domains:categorical and numerical
- For classification – label (target) class variable is always categorical
- Predictors/input variables can be either categorical or numerical

What all can we do with a dataset?

Broad tasks

- ✗ Visualize
- ➔ Unearth patterns & relationships
  - ✗ Group them, descriptive analytics
  - ✓ Categorize, predict, classify
  - ✗ Rank/Order
  - ✗ Associations

M/L: Goals and types

Machine Learning Goals

- unearth “hidden” structures (unsupervised)
- predict/forecast (supervised)

Classifier – Lingo

web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf

1 Introduction

Consider the standard supervised learning problem. A learning program is given training examples of the form $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ for some unknown function $y = f(\mathbf{x})$. The $\mathbf{x}_i$ values are typically vectors of the form $\langle x_{i,1}, x_{i,2}, \ldots, x_{i,n} \rangle$ whose components are discrete- or real-valued such as height, weight, color, age, and so on. These are also called the *features* of $\mathbf{x}_i$. Let us use the notation x_{ij} to refer to the j-th feature of $\mathbf{x}_i$. In some situations, we will drop the i subscript when it is implied by the context.

The y values are typically drawn from a discrete set of classes $\{1, \ldots, K\}$ in the case of *classification* or from the real line in the case of *regression*. In this chapter, we will consider only classification. The training examples may be corrupted by some random noise.

Given a set S of training examples, a learning algorithm outputs a *classifier*. The classifier is an hypothesis about the true function f. Given new $\mathbf{x}$ values, it predicts the corresponding y values. I will denote classifiers by $h_1, \ldots, h_L$.

ML Workflow

https://mapr.com/ebooks/spark/08-recommendation-engine-spark.html

ISL | Statistics | How to Stop Skype Fr | Attributes: Defining C | Fundry & Yesco - Firm

Typical Machine Learning Workflow

A typical machine learning workflow is shown below.

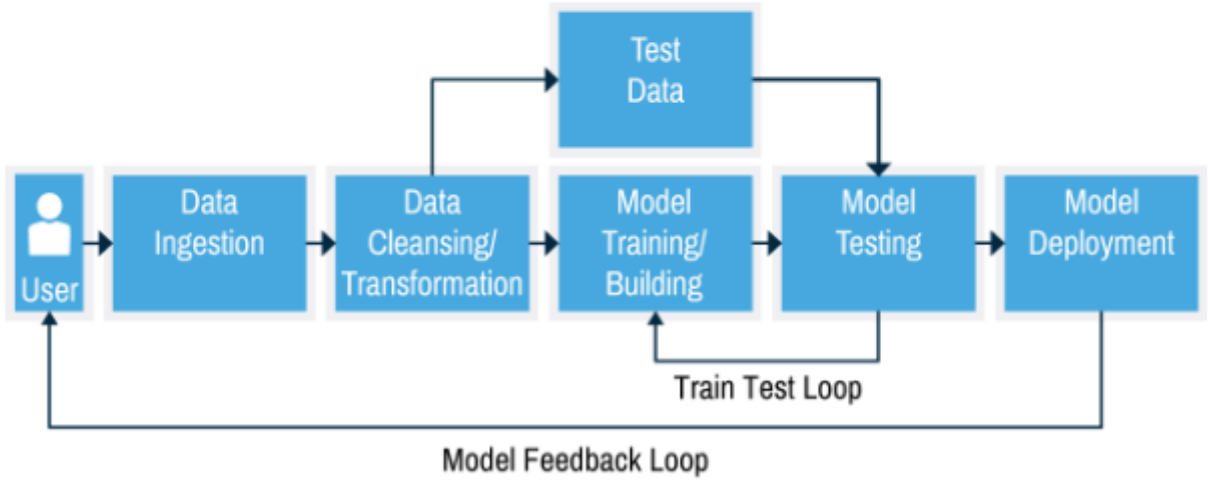
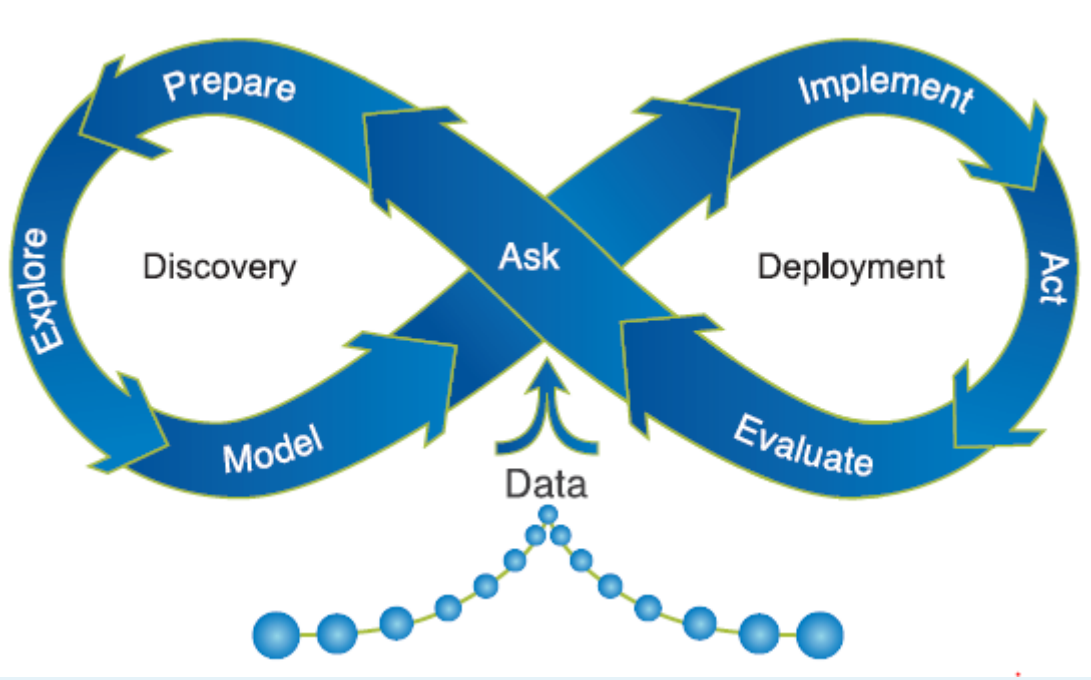


Figure 12. Maching Learning Workflow

Process is – Iterative



This process applies to learning, data mining and M/L and it is iterative – not a single-pass sequence of steps.

https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/data-mining-from-a-z-104937.pdf

2 Phase Learning

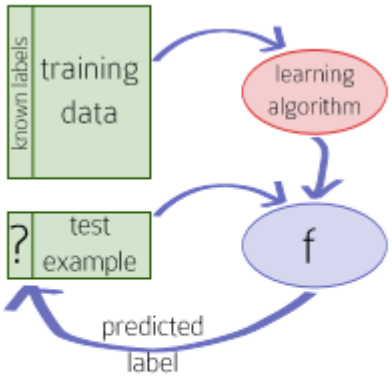


Figure 1.1: The general supervised approach to machine learning: a learning algorithm reads in training data and computes a learned function f. This function can then automatically label future text examples.

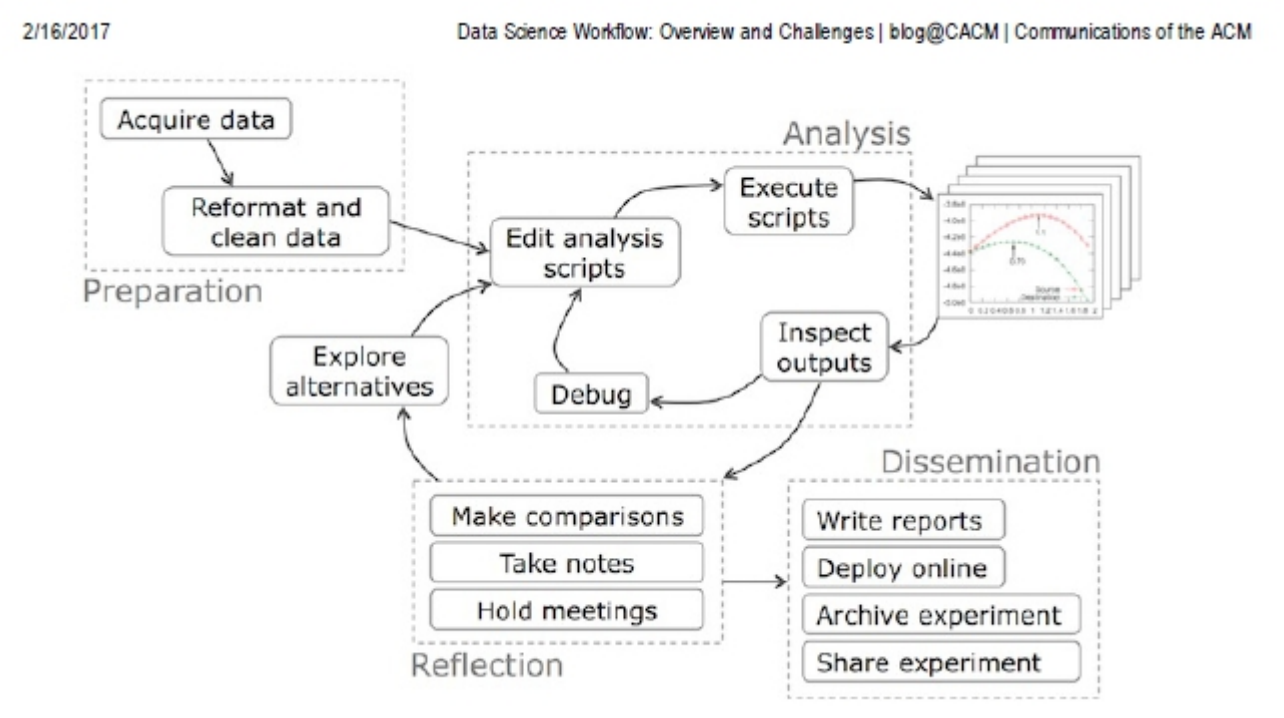
The general framework of induction.
We are given data on which our algorithm is expected to learn to assign a class.label to new data.
We cannot deploy without knowing how the classifier will perform when presented with never seen before data in production.
So we split the given data into two disjoint partitions training set and
test set. The algorithm is trained on the training set.induces a function f that will map a new example to a class.
We use the test set to evaluate our algorithm to determine how the algorithm will perform over never seen before data. The performance over the test data is a good indication of how the classifier will perform when new data arrives, post deployment.
Ability to learn from a given set of observations and reason when new data is presented is known as Generalization.

The goal of inductive machine learning is to take some training data and use it to induce a function f. This function f will be evaluated on the test data. The machine learning algorithm has succeeded if its performance on the test data is high.
Classification is about Generalization.

s
What question comes to your mind?

ciml.info/dl/v0_99/ciml-v0_99-ch01.pdf

Closer look inside



http://cacm.acm.org/blogs/blog-cacm/169199datascienceworkflowoverviewandchallenges/fulltext

Process Discipline

Process Discipline
→ Process Script
→ ***Repeatable/Reproducible***
→Different compositions of training set will result in different learners
→Consider two bags A and B.
→In A 6 oranges and 6 apples.
→In B 2 oranges and 10 apples
→What we learn from A and B
→are different.
→But we want repeatability and reproducibility.
→To achieve that, we always set a seed and then randomly partition the data into training and test set.
→It may be instructive to ensure distribution of classes are similar between the training and test sets.

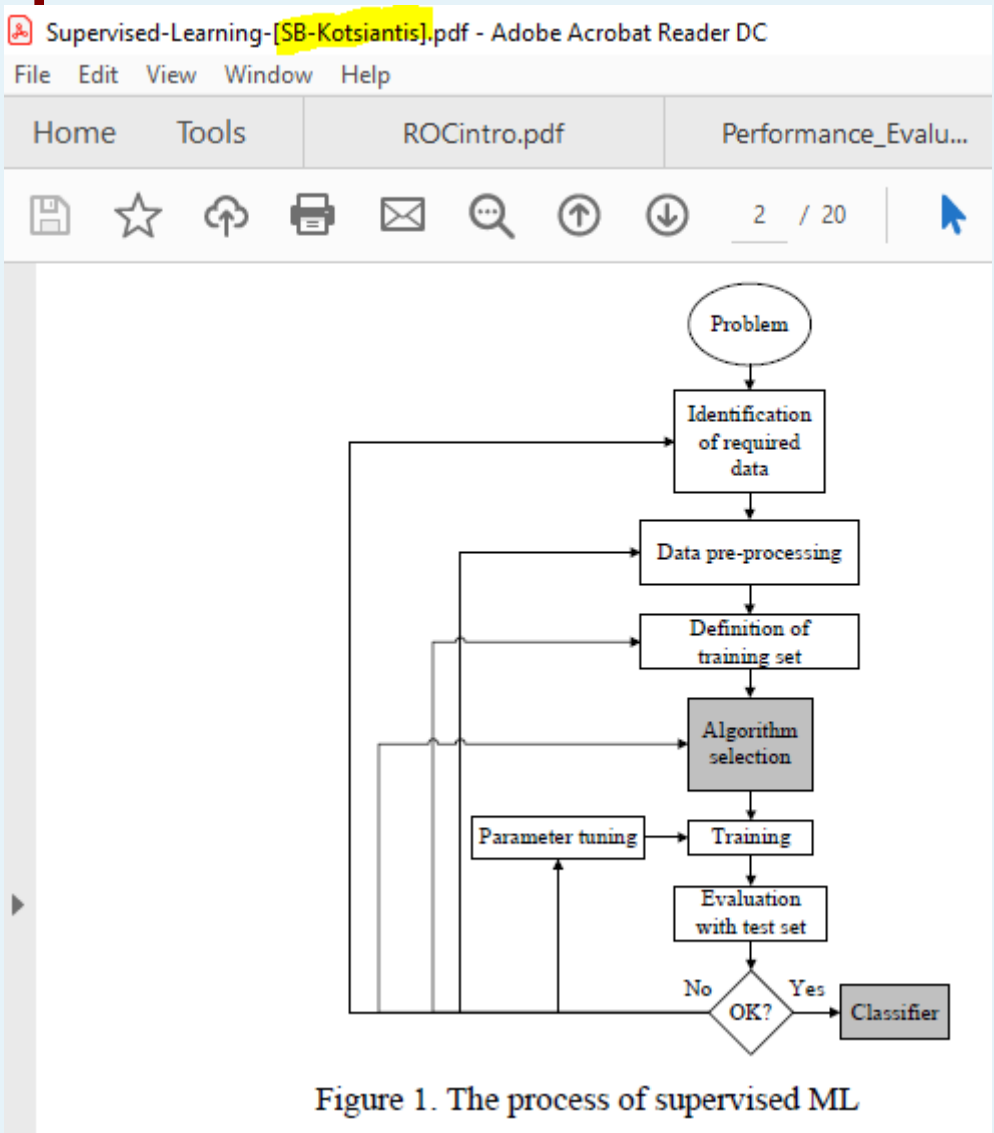


Figure 1. The process of supervised ML

Performance: A Formalism

- How can we assess the classifier is performant over never seen before data?
- But, *what is performance for a classifier?*

- *rMSE is not useful*

Actual	Predicted	
Pass	Pass	TP
Pass	Fail	FN
Fail	Pass	FP
Fail	Fail	TN

- The performance of the learning algorithm should be measured on unseen "test" data.
- The way in which we measure performance should depend on the problem we are trying to solve.
- There should be a strong relationship between the data that our algorithm sees at training time and the data it sees at test time.

In order to accomplish this, let's assume that someone gives us a **loss function**, $\ell(\cdot, \cdot)$, of two arguments. The job of ℓ is to tell us how "bad" a system's prediction is in comparison to the truth. In particular, if y is the truth and $\hat{y}$ is the system's prediction, then $\ell(y, \hat{y})$ is a measure of error.

WHAT

WHY

HOW

Never seen before data – Generalize

Confusion Matrix

and negative

buys_computer = *no*. Suppose we use our classifier on a test set of labeled tuples. P is the number of positive tuples and N is the number of negative tuples. For each tuple, we compare the classifier's class label prediction with the tuple's known class label.

There are four additional terms we need to know that are the "building blocks" used in computing many evaluation measures. Understanding them will make it easy to grasp the meaning of the various measures.

- **True positives** (TP): These refer to the positive tuples that were correctly labeled by the classifier. Let TP be the number of true positives.
- **True negatives** (TN): These are the negative tuples that were correctly labeled by the classifier. Let TN be the number of true negatives.
- **False positives** (FP): These are the negative tuples that were incorrectly labeled as positive (e.g., tuples of class *buys_computer* = *no* for which the classifier predicted *buys_computer* = *yes*). Let FP be the number of false positives.
- **False negatives** (FN): These are the positive tuples that were mislabeled as negative (e.g., tuples of class *buys_computer* = *yes* for which the classifier predicted *buys_computer* = *no*). Let FN be the number of false negatives.

These terms are summarized in the **confusion matrix** of Figure 8.14.

The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of

Confusion Matrix

and negative

buys_computer = *no*. Suppose we use our classifier on a test set of labeled tuples. P is the number of positive tuples and N is the number of negative tuples. For each tuple, we compare the classifier's class label prediction with the tuple's known class label.

There are four additional terms we need to know that are the "building blocks" used in computing many evaluation measures. Understanding them will make it easy to grasp the meaning of the various measures.

- **True positives** (TP): These refer to the positive tuples that were correctly labeled by the classifier. Let TP be the number of true positives.
- **True negatives** (TN): These are the negative tuples that were correctly labeled by the classifier. Let TN be the number of true negatives.
- **False positives** (FP): These are the negative tuples that were incorrectly labeled as positive (e.g., tuples of class *buys_computer* = *no* for which the classifier predicted *buys_computer* = *yes*). Let FP be the number of false positives.
- **False negatives** (FN): These are the positive tuples that were mislabeled as negative (e.g., tuples of class *buys_computer* = *yes* for which the classifier predicted *buys_computer* = *no*). Let FN be the number of false negatives.

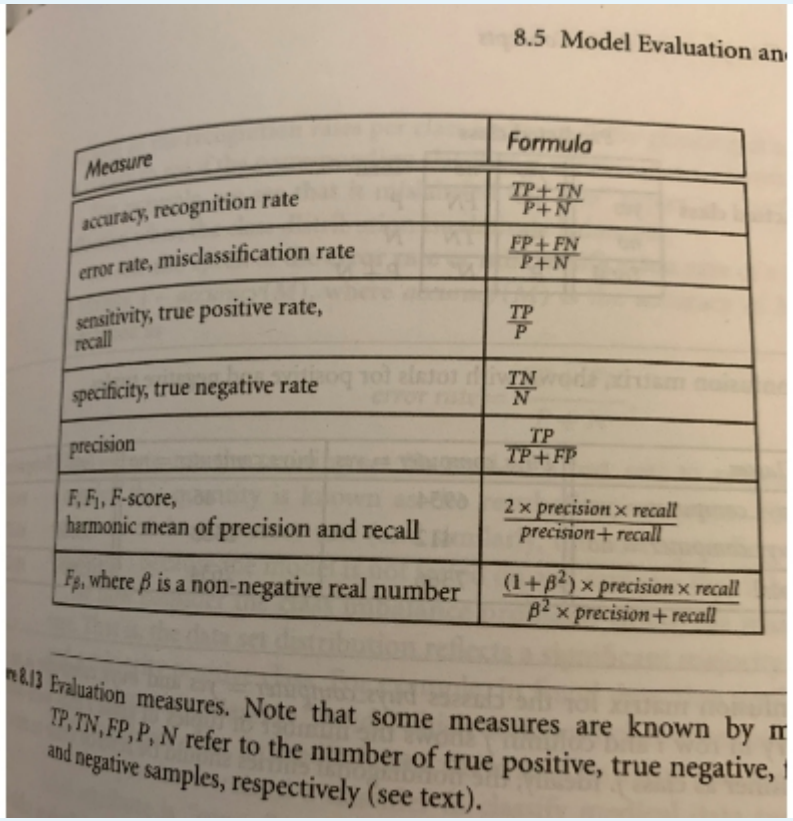
These terms are summarized in the **confusion matrix** of Figure 8.14.

The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of

Classifier Evaluation Metrics

Metrics

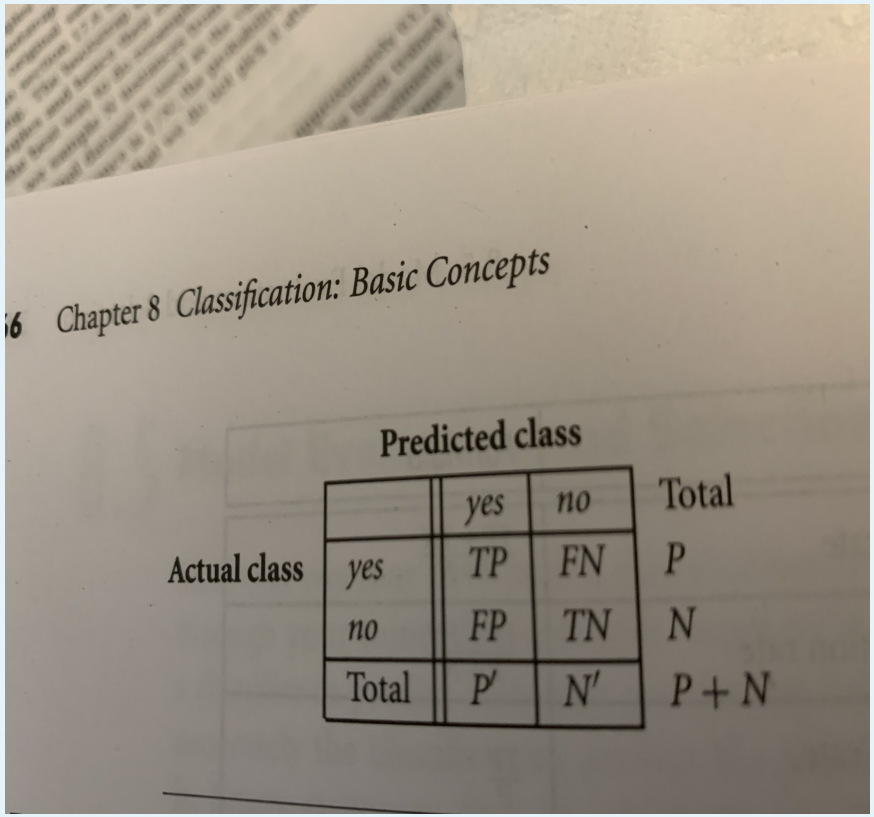
Confusion Matrix



8.5 Model Evaluation an

Measure	Formula
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
F, F_1, F-score, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β, where β is a non-negative real number	$\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

8.13 Evaluation measures. Note that some measures are known by m
TP, TN, FP, P, N refer to the number of true positive, true negative,
and negative samples, respectively (see text).



Chapter 8 Classification: Basic Concepts

		Predicted class: yes	Predicted class: no	Total
Actual class	yes	TP	FN	P
	no	FP	TN	N
	Total	P′	N′	P + N

Introduction to Machine Learning – Ethem Alpaydin

Other Evaluation Metrics

Table 19.3 Type I error, type II error, and power of a test.

	Decision	
Truth	Fail to reject	Reject
True	Correct	Type I error
False	Type II error	Correct (power)

alpha

Beta

ROC is a visual tool to evaluate classifiers.

AUC, Area Under the curve
Is numerical measure

7. Area under an ROC curve (AUC)

An ROC curve is a two-dimensional depiction of classifier performance. To compare classifiers we may want to reduce ROC performance to a single scalar value representing expected performance. A common method is to calculate the area under the ROC curve, abbreviated **AUC** (Bradley, 1997; Hanley and McNeil, 1982). Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1.0. However, because random guessing produces the diagonal line between (0,0) and (1,1), which has an area of 0.5, no realistic classifier should have an AUC less than 0.5.

The AUC has an important statistical property: the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. This is equivalent to the Wilcoxon test of ranks (Hanley and McNeil, 1982). The AUC is also closely related to the Gini coefficient (Breiman et al., 1984), which is twice the area between the diagonal and the ROC curve. Hand and Till (2001) point out that $\text{Gini} + 1 = 2 \times \text{AUC}$.

Fig. 8a shows the areas under two ROC curves, A and B. Classifier B has greater area and therefore better average performance. Fig. 8b shows the area under the curve of a binary classifier A and a scoring classifier B. Classifier A represents the performance of B when B is used with a single, fixed threshold. Though the performance of the two is equal at the fixed point (A's threshold), A's performance becomes inferior to B further from this point.

It is possible for a high-AUC classifier to perform worse in a specific region of ROC space than a low-AUC classifier. Fig. 8a shows an example of this: classifier B is generally better than A except at FPrate > 0.6 where A has a

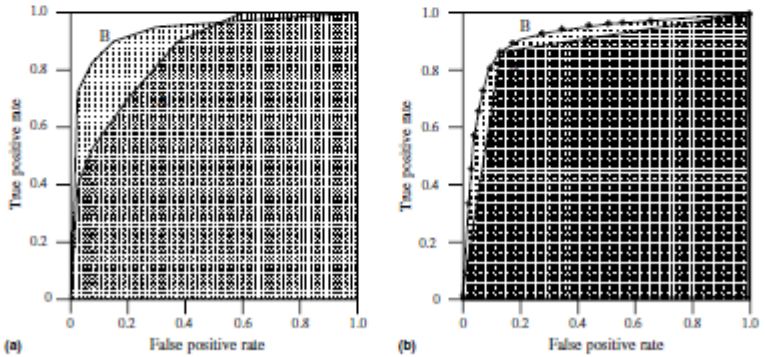


Fig. 8. Two ROC graphs. The graph on the left shows the area under two ROC curves. The graph on the right shows the area under the curves of a discrete classifier (A) and a probabilistic classifier (B).

An introduction to ROC analysis
Tom Fawcet

Status

That completes a grand tour of many essential concepts relating to Classification, classifiers and their evaluation.

We will now apply these concepts to classification exercise Using several different classifiers.

We will analyze heart.csv and ecoli.csv.

Our Routine

1. Load data
2. Preliminary EDA
3. Training/Testing Sets
4. Train – determine underfitting
5. Predict over Test set
6. Rule out overfitting
7. ROC and AUC meaures