# Machine Learning Workshops

## Introduction

Mert Kosan
Department of Computer Science
University of California, Santa Barbara

# Workshop Outline

- Introduction to Machine Learning (Thurs. 9.00 - 10.45)
  - Motivation, History, Related Disciplines and Examples
  - Kinds of Machine Learning
  - Learning: Key Steps
  - General Pipeline of Machine Learning Projects
- Supervised Setting (Thurs. 13.30 - 15.15)
  - Regression
  - Classification
- Unsupervised Setting (Fri. 13.30 - 15.15)
  - Clustering
  - Dimensionality Reduction
  - Deep Learning for Unsupervised Learning
- Practical Examples (Fri. 15.15 - 17.00)
  - Iris, Swiss roll, MNIST etc.

# Introduction to Machine Learning Outline

- The Motivation, Related Disciplines, Examples, and History
- Major Paradigms/Kinds of ML
  - Rote Learning
  - Supervised Learning (Regression, Classification)
  - Unsupervised Learning (Clustering)
  - Reinforcement Learning
- Learning: Key Steps
  - Data and Assumptions, Representation, Method and Estimation, Evaluation
- General pipeline for Machine Learning Project
  - Train, test, and validation
  - Cross-Validation
  - Design Cycle
  - Overfitting and Model complexity

# Introduction to Machine Learning Outline

- **The Motivation, Related Disciplines, Examples, and History**
- Major Paradigms/Kinds of ML
  - Rote Learning
  - Supervised Learning (Regression, Classification)
  - Unsupervised Learning (Clustering)
  - Reinforcement Learning
- Learning: Key Steps
  - Data and Assumptions, Representation, Method and Estimation, Evaluation
- General pipeline for Machine Learning Project
  - Train, test, and validation
  - Cross-Validation
  - Design Cycle
  - Overfitting and Model complexity

# What is learning?

Some definitions from famous people in CS/AI history:

- “Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time.” –Herbert Simon
- “Learning is any process by which a system improves performance from experience.” –Herbert Simon
- “Learning is constructing or modifying representations of what is being experienced.” –Ryszard Michalski
- “Learning is making useful changes in our minds.” –Marvin Minsky

Keywords: Experience, Enhancement

# Defining the Learning Task

“A computer program is said to learn from experience E with some class of tasks T and performance measure P…” -Tom M. Mitchell

Examples:

- Checkers
  - T: Playing checkers
  - P: Percentage of games won against an arbitrary opponent
  - E: Playing practice games against itself
- Hand-written words
  - T: Recognizing hand-written words
  - P: Percentage of words correctly classified
  - E: Database of human-labeled images of handwritten words

# Defining the Learning Task

“A computer program is said to learn from experience E with some class of tasks T and performance measure P…” -Tom M. Mitchell

Examples:

- Driving
  - T: Driving on four-lane highways using vision sensors
  - P: Average distance traveled before a human-judged error
  - E: A sequence of images and steering commands recorded while observing a human driver.
- Spam or not
  - T: Categorize email messages as spam or legitimate.
  - P: Percentage of email messages correctly classified.
  - E: Database of emails, some with human-given labels

# Why learning?

- Build software agents that can adapt to their users or to other software agents or to changing environments
  - Mars robot
- Develop systems that are too difficult/expensive to construct manually because they require specific detailed skills or knowledge tuned to a specific task
  - Large, complex AI systems cannot be completely derived by hand and require dynamic updating to incorporate new information.
- Discover new things that were previously unknown to humans
  - Examples: data mining, scientific discovery

# Related Disciplines

The following are close disciplines:

- Artificial Intelligence: Machine learning deals with the learning part of AI.
- Pattern Recognition: Concentrates more on “tools” rather than theory.
- Data Mining: More specific about discovery.

The following are useful in machine learning techniques or may give insights:

- Probability and Statistics
- Information theory
- Psychology (developmental, cognitive)
- Neurobiology
- Linguistics
- Philosophy

# History

- 1950s:
  - Samuel’s checker player
- 1960s:
  - Neural networks: Perceptron
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Expert systems
- 1980s:
  - Resurgence of neural networks (connectionism, backpropagation)
  - Utility Theory

# History

- 1990s:
  - Data mining
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
- 2000s:
  - Kernel methods
    - Support vector machines
  - Graphical models
  - Statistical relational learning
  - Deep learning
    - Deep Blue (chess-IBM), AlphaGO (Go-Deepmind)

# Introduction to Machine Learning Outline

- The Motivation, Related Disciplines, Examples, and History
- **Major Paradigms/Kinds of ML**
  - **Rote Learning**
  - **Supervised Learning (Regression, Classification)**
  - **Unsupervised Learning (Clustering)**
  - **Reinforcement Learning**
- Learning: Key Steps
  - Data and Assumptions, Representation, Method and Estimation, Evaluation
- General pipeline for Machine Learning Project
  - Train, test, and validation
  - Cross-Validation
  - Design Cycle
  - Overfitting and Model complexity

# Major kinds of Machine Learning

- Rote learning: “Learn by memorization”
  - Employed by first machine learning systems, in 1950s
    - Samuel’s Checkers program
- Supervised learning – Use specific examples to reach general conclusions or extract general rules
  - Classification / Concept learning
  - Regression
- Unsupervised learning – Unsupervised identification of natural groups in data
  - Clustering
- Reinforcement learning– Feedback (positive or negative reward) given at the end of a sequence of steps

# Rote Learning is Limited

- Memorize I/O pairs and perform exact matching with new inputs.
- If a computer has not seen the precise case before, it cannot apply its experience.
- We want computers to “generalize” from prior experience
  - Generalization is the most important factor in learning.

# The inductive learning problem

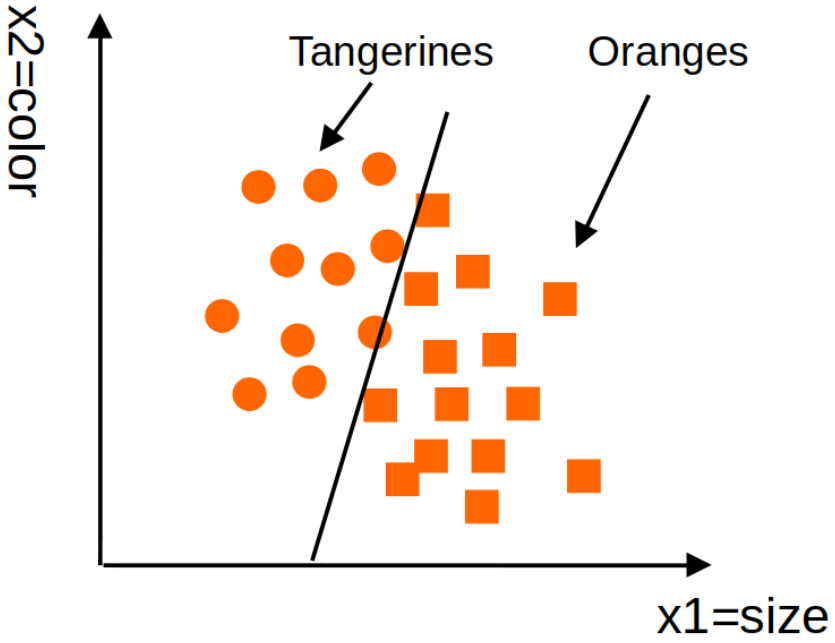
- Extrapolate from a given set of examples to make accurate predictions about future examples

- Supervised versus unsupervised learning
  - Learn an unknown function f(X) = Y, where X is an input example and Y is the desired output.
  - **Supervised learning** implies we are given a **training set** of (X, Y) pairs by a “teacher”
  - **Unsupervised learning** means we are only given the Xs.
  - **Semi-supervised learning:** mostly unlabelled data

# Types of Supervised Learning

**Classification**

- We are given the label of the training objects: {(x1,x2,y=T/O)}
- We are interested in classifying future objects: (x1',x2') with the correct label I.e. Find y' for given (x1',x2').

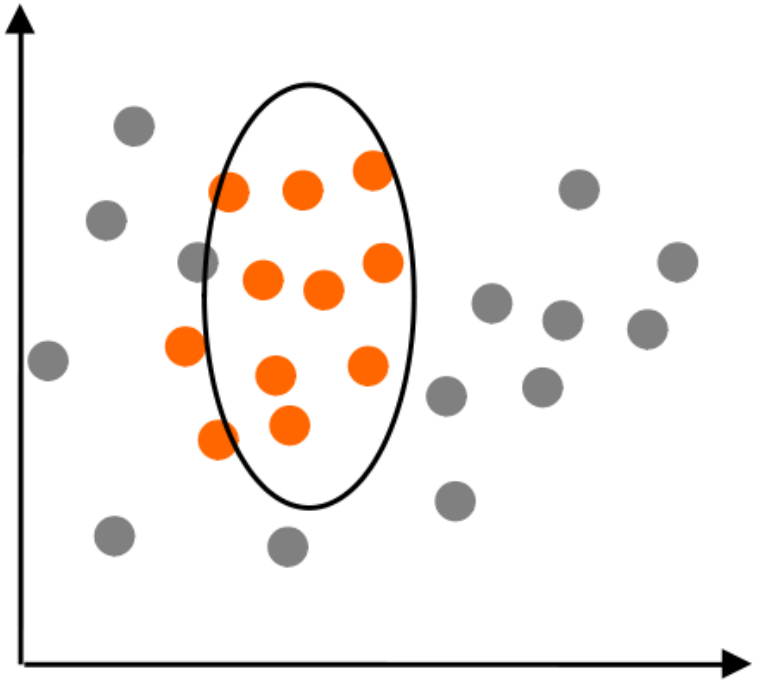


# Types of Supervised Learning

**Concept Learning:**

- We are given positive and negative samples for the concept we want to learn (e.g.Tangerine): {(x1,x2,y=+/-)}

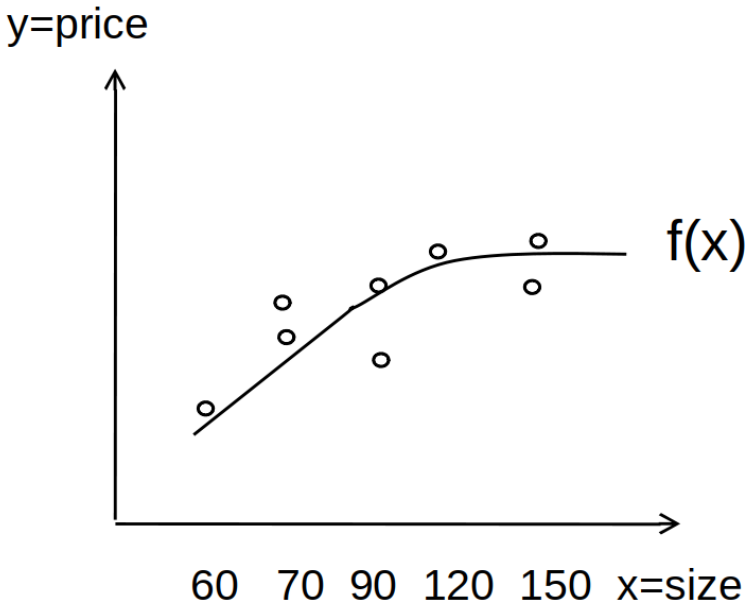
- We are interested in classifying future objects as member of the class (or positive example for the concept) or not. I.e. Answer +/- for given (x1',x2').



# Types of Supervised Learning

**Regression:**

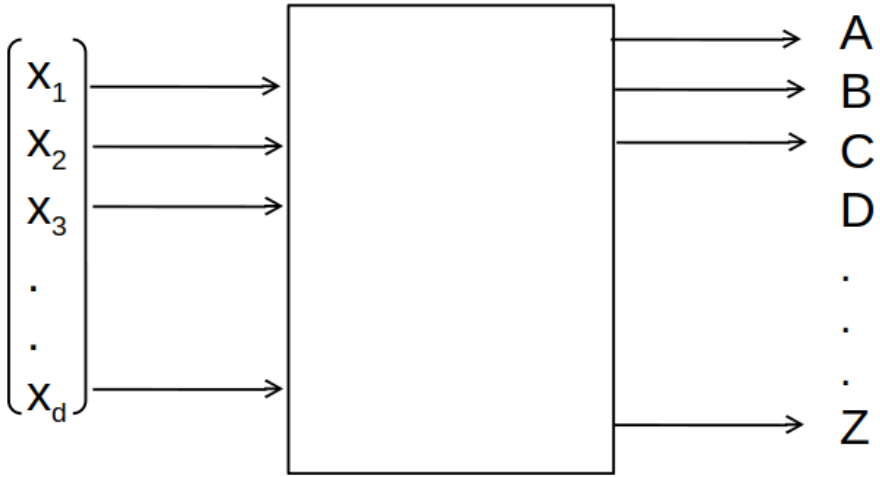
- Target function is **continuous** rather than class membership.
- For example, you have some the selling prices of houses. You will learn function f(x) which tells the price of the house given its sq-mt.
- The problem is more meaningful and challenging if you imagine several input parameters, resulting in a multi-dimensional input space.



# Classification

- Assign object/event to one of a given finite set of categories.
  - Medical diagnosis
  - Credit card applications or transactions
  - Fraud detection in e-commerce
  - Spam filtering in email
  - Recommended books, movies, music
  - Financial investments
  - Spoken words
  - Handwritten letters

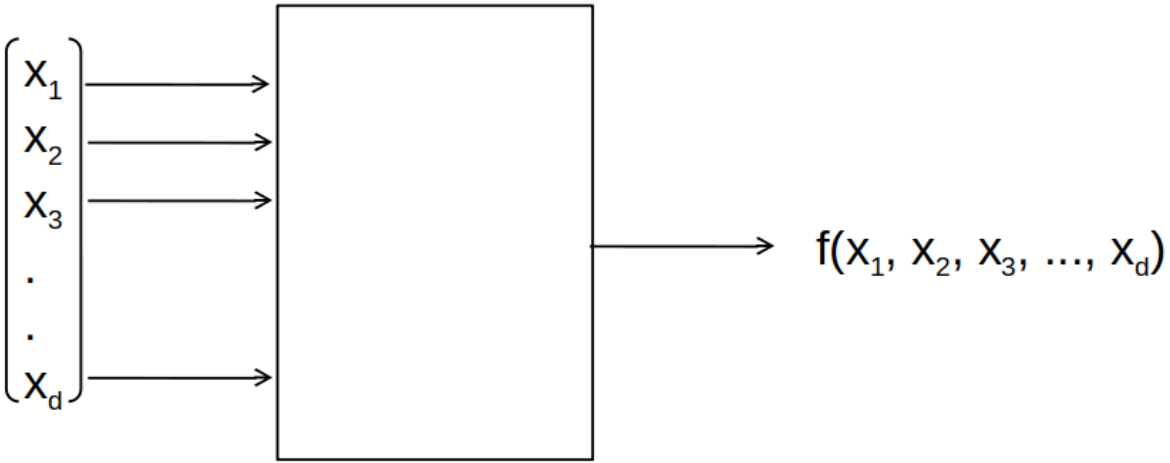
# Classification System




Input Classifier Output

(Features or Feature Vector)

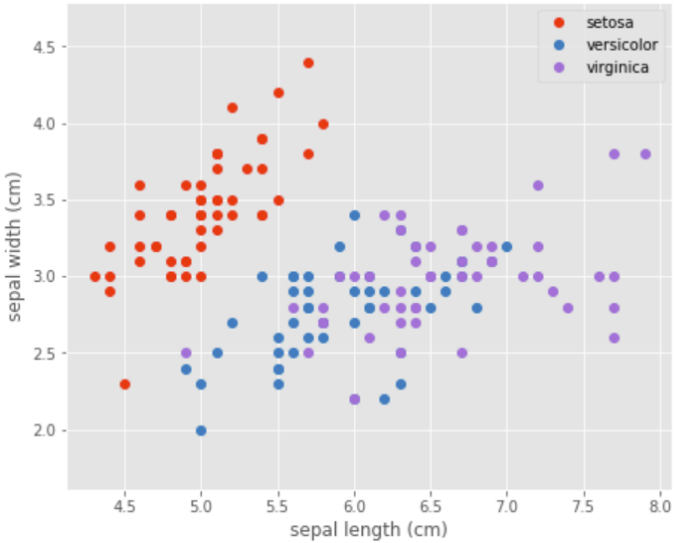
# Regression System

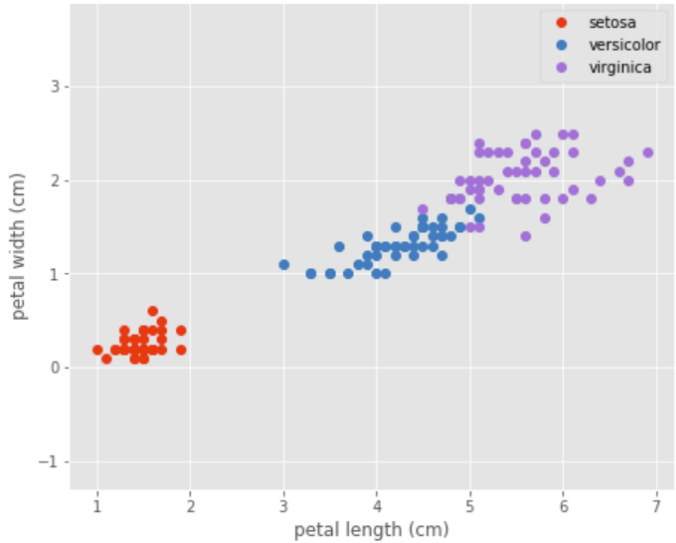



Input Classifier Output

(Features or Feature Vector)

# Iris




- The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper “The use of multiple measurements in taxonomic problems” as an example of linear discriminant analysis. [Wikipedia, Iris flower data set]

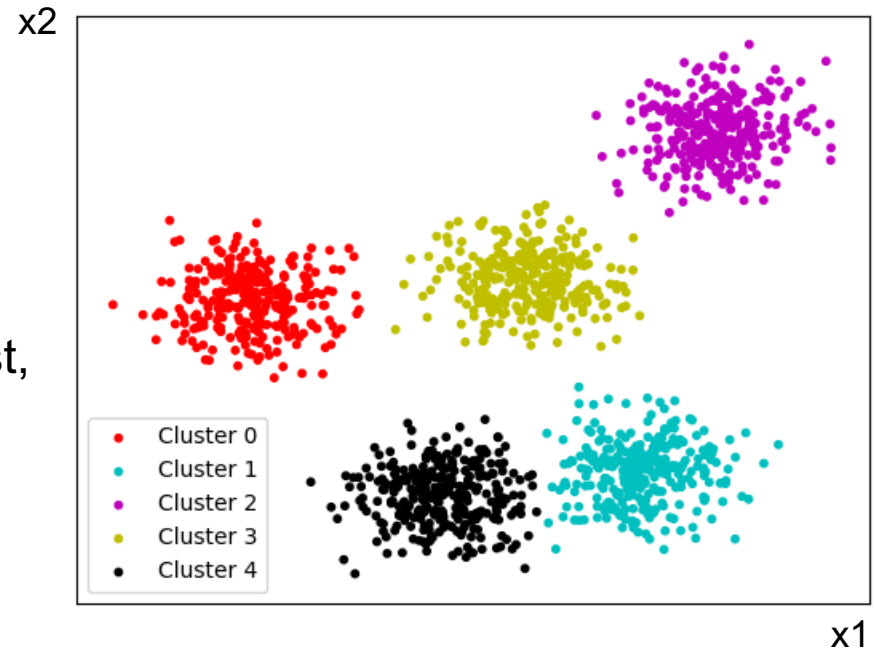
# MNIST database

- The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. [Wikipedia, MNIST database]

# Types of Unsupervised Learning

**Clustering:**

- We are given training objects without labels: {(x1,x2,y=?)}
- We are interested in clustering the training objects and labeling them first, then classify future objects: (x1',x2') with the correct label I.e. Find y' for given (x1',x2').

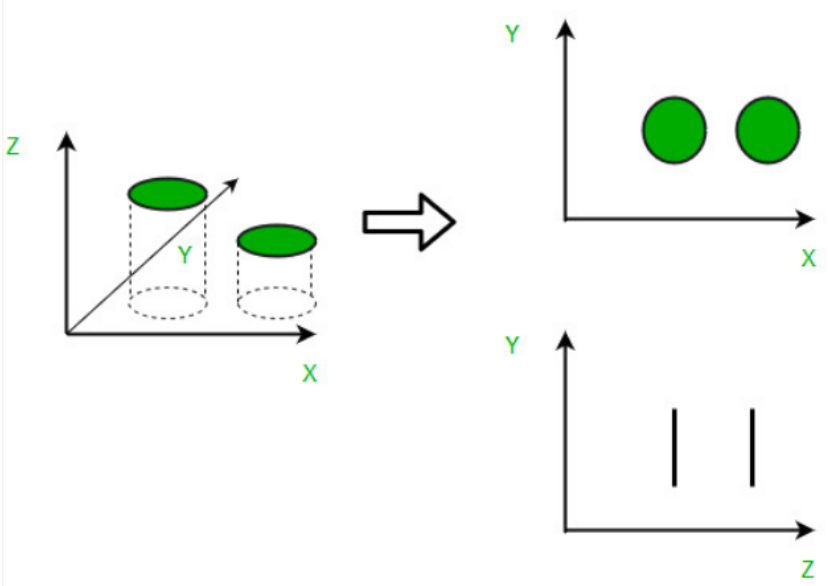



https://www.imperva.com/blog/clustering-and-dimensionality-reduction-understanding-the-magic-behind-machine-learning/

# Types of Unsupervised Learning

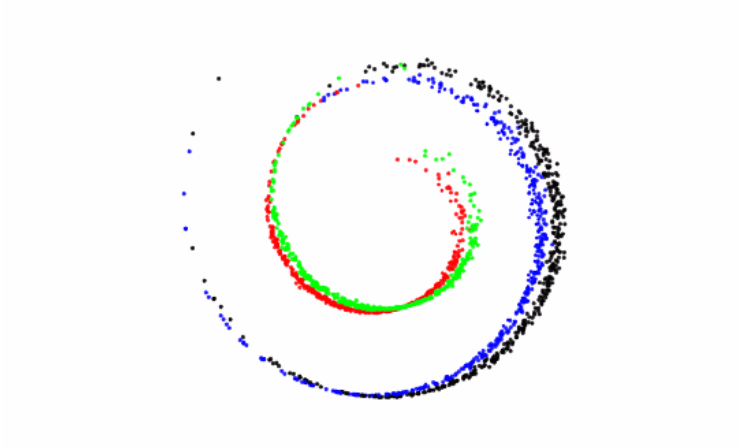
**Dimensionality Reduction:**

- Curse of Dimensionality: Most of time, large of number of features won't help.
- We are interested in decreasing the number of features. The aim is to find the subset of features that help most.




https://www.geeksforgeeks.org/dimensionality-reduction/

# Swiss Roll

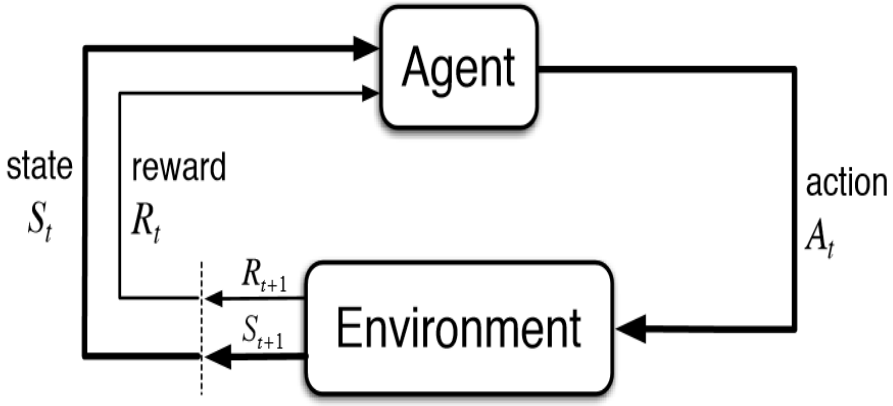


- Swiss Roll
  - The original data was created by randomly sampling from a Gaussian Mixture Model with centers/means at (7.5,7.5), (7.5,12.5), (12.5,7.5) and (12.5,12.5).
  - Lies intrinsically in 2D manifold, with 4 clusters. Perfect for testing dimensionality reduction and clustering algorithms.

# Reinforcement Learning

Problems involving an interaction between an **agent** and an **environment**, which provides **reward** signal.

**Goal:** Learning how to take actions in order to maximize reward.




https://thegradient.pub/why-rl-is-flawed/

Example - Atari Games:

- Objective: Get the highest score.
- State: Game state.
- Action: Game controls.
- Reward: Score increase/decrease each step.

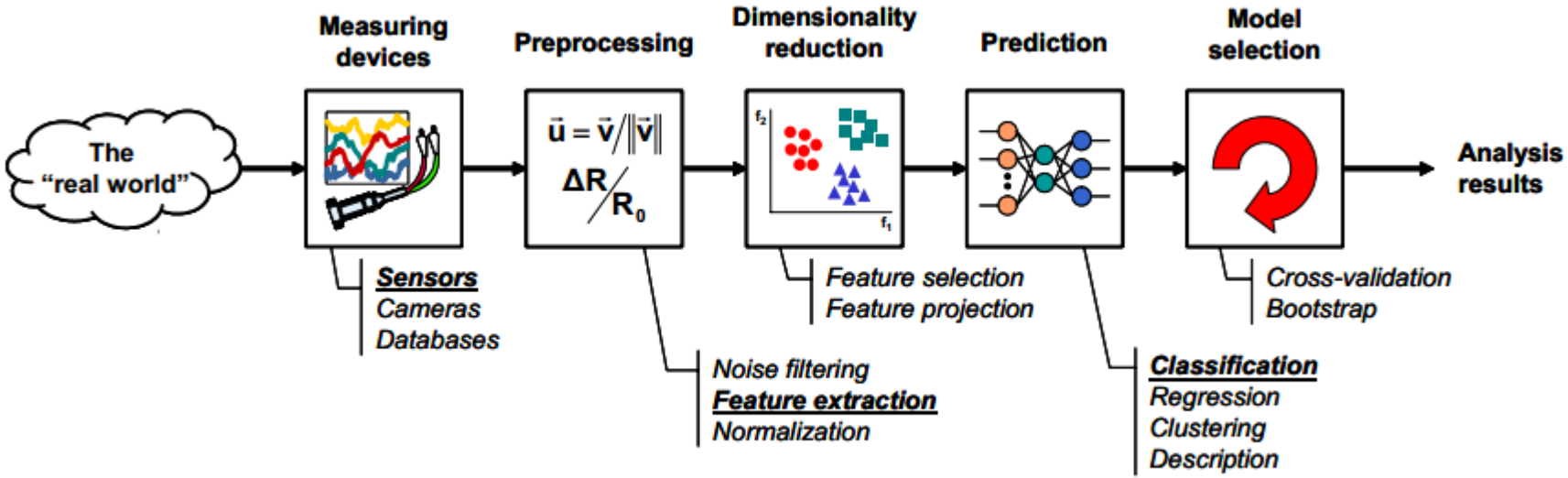
# Introduction to Machine Learning Outline

- The Motivation, Related Disciplines, Examples, and History
- Major Paradigms/Kinds of ML
  - Rote Learning
  - Supervised Learning (Regression, Classification)
  - Unsupervised Learning (Clustering)
  - Reinforcement Learning
- **Learning: Key Steps**
  - **Data and Assumptions, Representation, Method and Estimation, Evaluation**
- General pipeline for Machine Learning Project
  - Train, test, and validation
  - Cross-Validation
  - Design Cycle
  - Overfitting and Model complexity

# Learning: Key Steps

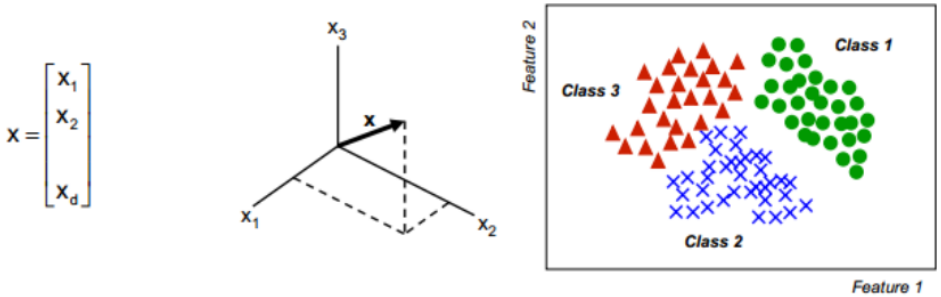
- Data and Assumptions
  - What data is available for the learning task?
  - What can we assume about the problem?
- Representation
  - How should we represent the examples to be classified?
- Method and Estimation
  - What are the possible hypotheses?
  - What learning algorithm to use to infer the most likely hypothesis?
- Evaluation
  - How well are we doing?

# Learning: Key Steps



# Feature

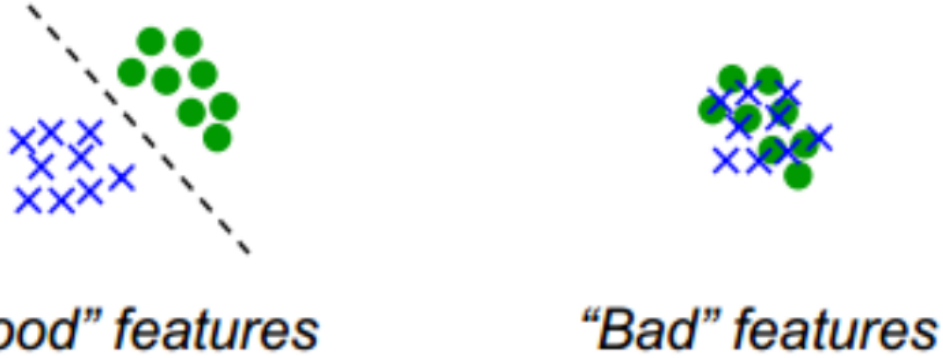
- Feature is any distinctive aspect, quality or characteristic.
  - They may be symbolic (i.e. color) or numeric (i.e. height)
- Definitions:
  - Feature vector: The combination of d features is represented as a d-dimensional column vector.
  - Feature space: The d-dimensional space defined by the feature vector.
  - Scatter plot: Objects are represented as points in the feature space.




Feature vector | Feature space (3D) | Scatter plot (2D)

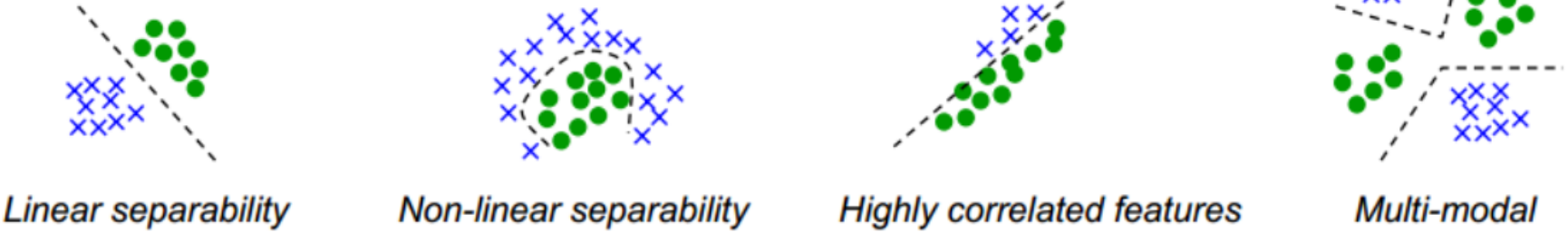
# Features and Patterns

- What makes a “good” feature vector?
  - The quality of a feature vector is related to its ability to discriminate examples from different classes.
    - Examples from the same class should have similar feature values.
    - Examples from different classes have different feature values.



# Features and Patterns

- More feature properties



# Evaluation of Learning Systems

- Experimental
  - Conduct controlled **cross-validation** experiments to compare various methods on a variety of benchmark datasets.
  - Gather data on their performance, e.g. **test accuracy, training-time, testing-time**…
  - Maybe even analyze differences for **statistical significance**.

- Theoretical
  - Analyze algorithms mathematically and prove theorems about their:
    - Ability to fit training data
    - Computational complexity
    - Sample complexity (number of training examples needed to learn an accurate function)

# Measuring Performance

- Performance of the learner can be measured in one of the following ways, as suitable for the application:
  - Accuracy
    - Number of mistakes (in classification problems)
    - Mean Squared Error (in regression problems)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

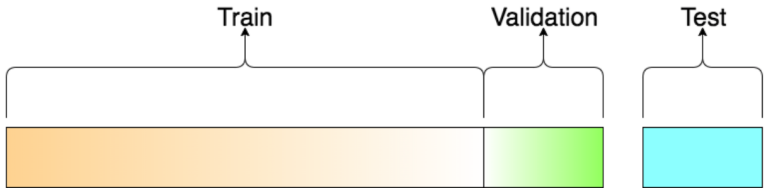
    - Loss functions (more general, taking into account different costs for different mistakes)
  - Solution quality (length, efficiency)
  - Speed of performance
  - …

# Introduction to Machine Learning Outline

- The Motivation, Related Disciplines, Examples, and History
- Major Paradigms/Kinds of ML
  - Rote Learning
  - Supervised Learning (Regression, Classification)
  - Unsupervised Learning (Clustering)
  - Reinforcement Learning
- Learning: Key Steps
  - Data and Assumptions, Representation, Method and Estimation, Evaluation
- **General pipeline for Machine Learning Project**
  - **Train, test, and validation**
  - **Cross-Validation**
  - **Design Cycle**
  - **Overfitting and Model complexity**

# Training, Validation, Test datasets

- Training dataset
  - The portion of dataset used for learning the model.
  - For example: to find f(x) in regression problem.
- Validation dataset
  - The portion of dataset used to tune the hyperparameters.
  - Hyperparameters: Parameters that has been set before the learning process starts.
  - Avoid overfitting ?
- Test dataset
  - The portion of dataset used to evaluate the model.

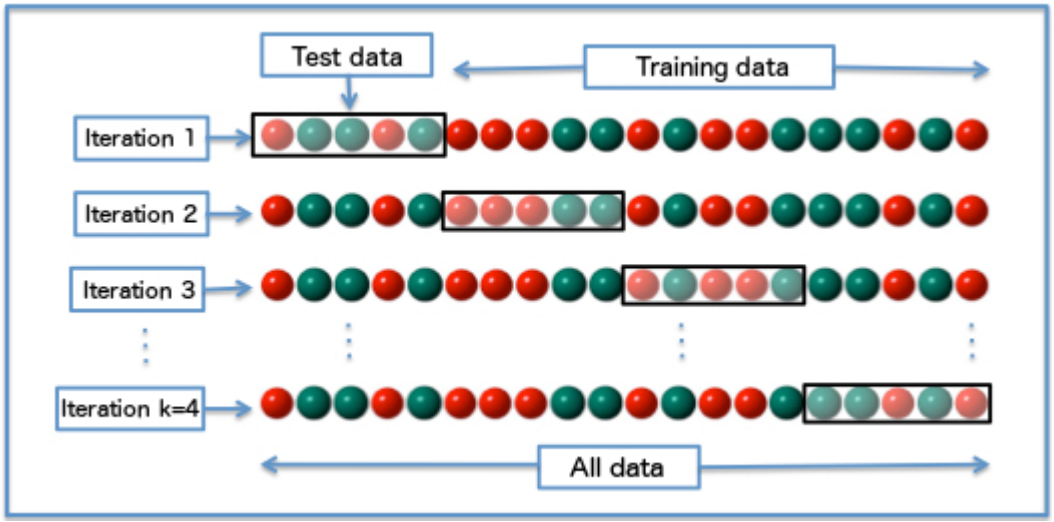


A visualisation of the splits

https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

# Cross-Validation

- You have some data to learn from and you need some test data to test your learnt system. A standard approach to evaluation is to use k-fold cross-validation.




https://en.wikipedia.org/wiki/Cross-validation_(statistics)

# Design Cycle

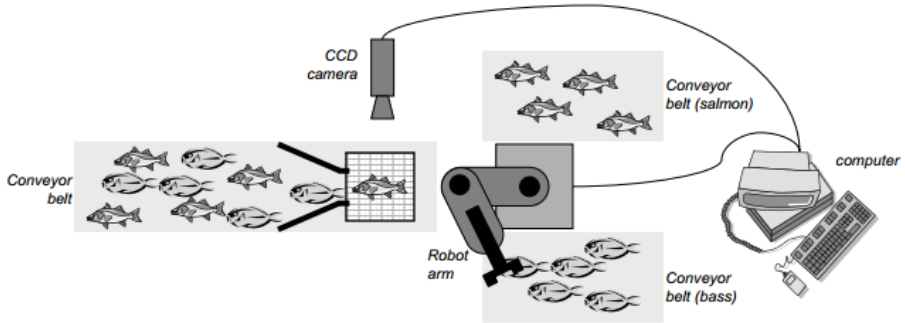
- The next 6 slides review the design cycle as whole process
  - From Gutierrez-Osuna, Texas A&M

# The pattern recognition design cycle (1)

- **Data collection**
  - Probably the most time-intensive component of a PR project
  - How many examples are enough?
- **Feature choice**
  - Critical to the success of the PR problem
    - "Garbage in, garbage out"
  - Requires basic prior knowledge
- **Model choice**
  - Statistical, neural and structural approaches
  - Parameter settings
- **Training**
  - Given a feature set and a "blank" model, adapt the model to explain the data
  - Supervised, unsupervised and reinforcement learning
- **Evaluation**
  - How well does the trained model do?
  - Overfitting vs. generalization

# The pattern recognition design cycle (2)

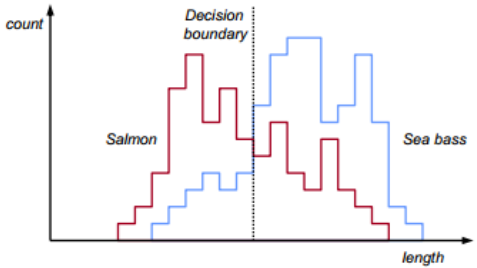
- **Consider the following scenario**
  - A fish processing plan wants to automate the process of sorting incoming fish according to species (salmon or sea bass)
  - The automation system consists of
    - a conveyor belt for incoming products
    - two conveyor belts for sorted products
    - a pick-and-place robotic arm
    - a vision system with an overhead CCD camera
    - a computer to analyze images and control the robot arm




*From [Duda, Hart and Stork, 2001]*

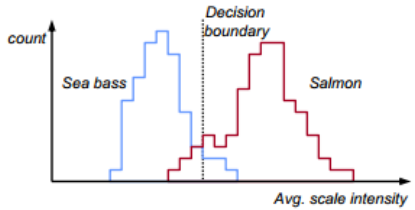
# The pattern recognition design cycle (3)

- **Sensor**
  - The vision system captures an image as a new fish enters the sorting area
- **Preprocessing**
  - Image processing algorithms
    - adjustments for average intensity levels
    - segmentation to separate fish from background
- **Feature Extraction**
  - Suppose we know that, on the average, sea bass is larger than salmon
    - From the segmented image we estimate the length of the fish
- **Classification**
  - Collect a set of examples from both species
  - Compute the distribution of lengths for both classes
  - Determine a decision boundary (threshold) that minimizes the classification error
  - We estimate the classifier's probability of error and obtain a discouraging result of 40%
  - **What do we do now?**

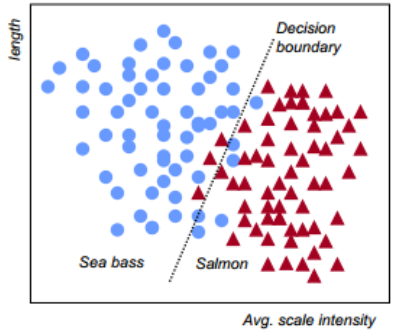


# The pattern recognition design cycle (4)

- **Improving the performance of our PR system**
  - Determined to achieve a recognition rate of 95%, we try a number of features
    - Width, Area, Position of the eyes w.r.t. mouth...
    - only to find out that these features contain no discriminatory information
  - Finally we find a "good" feature: average intensity of the scales

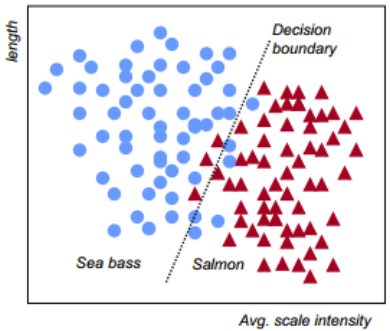


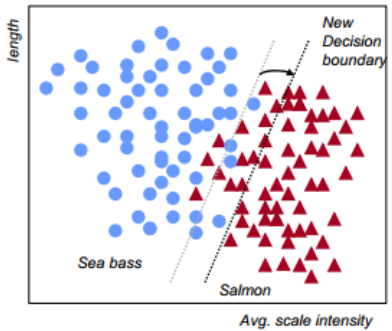

  - We combine "*length*" and "*average intensity of the scales*" to improve class separability
  - We compute a linear discriminant function to separate the two classes, and obtain a classification rate of 95.7%



# The pattern recognition design cycle (5)

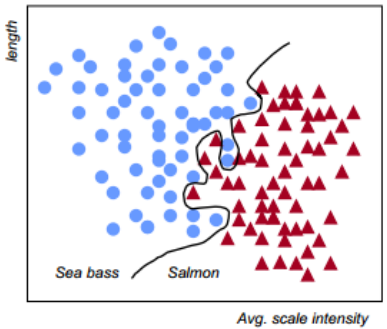
- **Cost Versus Classification rate**
  - Our linear classifier was designed to minimize the overall misclassification rate
  - Is this the best objective function for our fish processing plant?
    - The **cost** of misclassifying salmon as sea bass is that the end customer will occasionally find a tasty piece of salmon when he purchases sea bass
    - The **cost** of misclassifying sea bass as salmon is an end customer upset when he finds a piece of sea bass purchased at the price of salmon
  - Intuitively, we could adjust the decision boundary to minimize this cost function



# The pattern recognition design cycle (6)

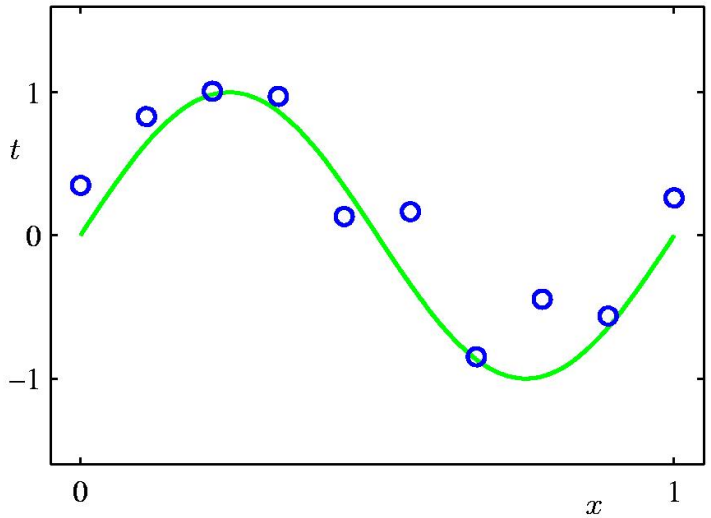
- **The issue of generalization**
  - The recognition rate of our linear classifier (95.7%) met the design specs, but we still think we can improve the performance of the system
    - We then design an artificial neural network with five hidden layers, a combination of logistic and hyperbolic tangent activation functions, train it with the Levenberg-Marquardt algorithm and obtain an impressive classification rate of 99.9975% with the following decision boundary




  - Satisfied with our classifier, we integrate the system and deploy it to the fish processing plant
    - After a few days, the plant manager calls to complain that the system is misclassifying an average of 25% of the fish
    - What went wrong?

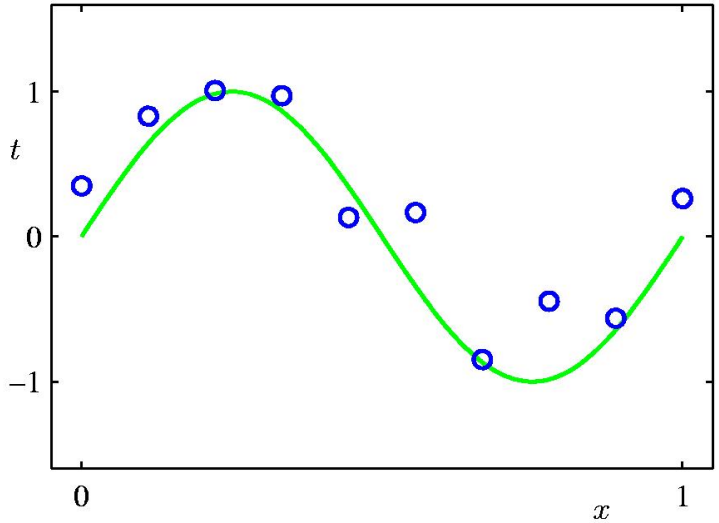
# Overfitting and Model Complexity

- Imagine that we have some training data(blue dots) and we want to learn the underlying function between the independent variable x and the target values t.



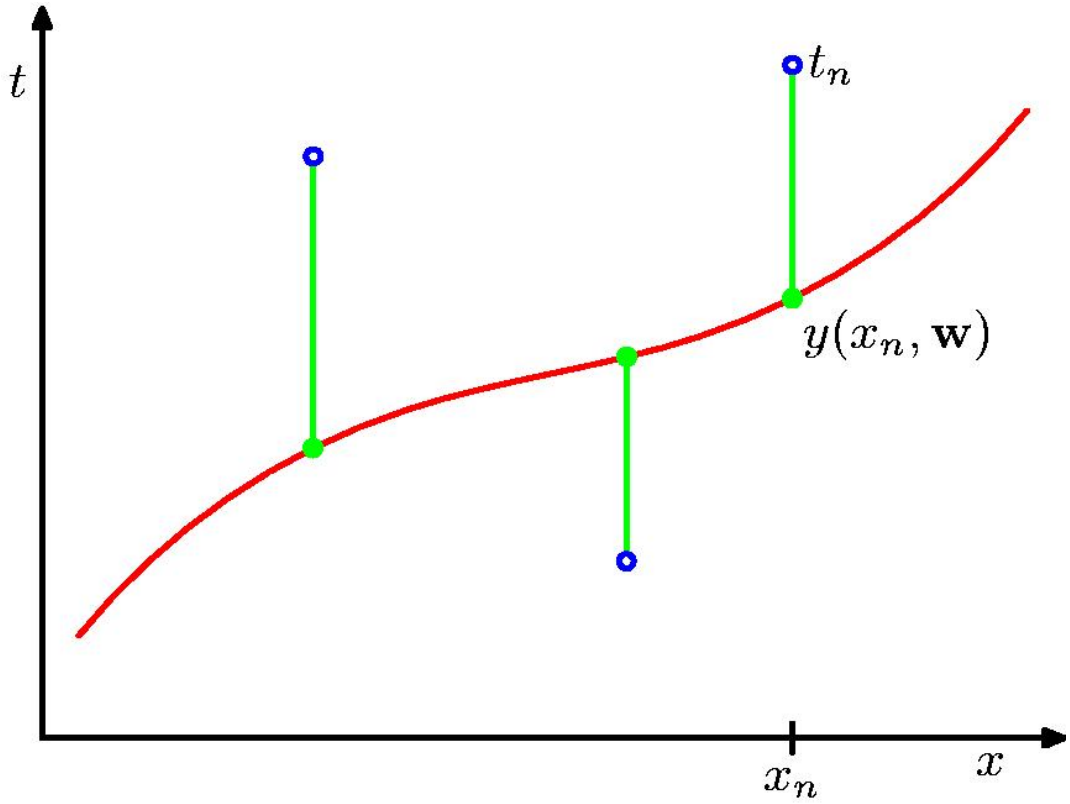
# Overfitting and Model Complexity

- We can fit polynomials in varying degrees: lines to higher degree polynomials.
- Higher degrees make the polynomial very capable to bend/flex to match the data as it has many parameters to change/adapt.



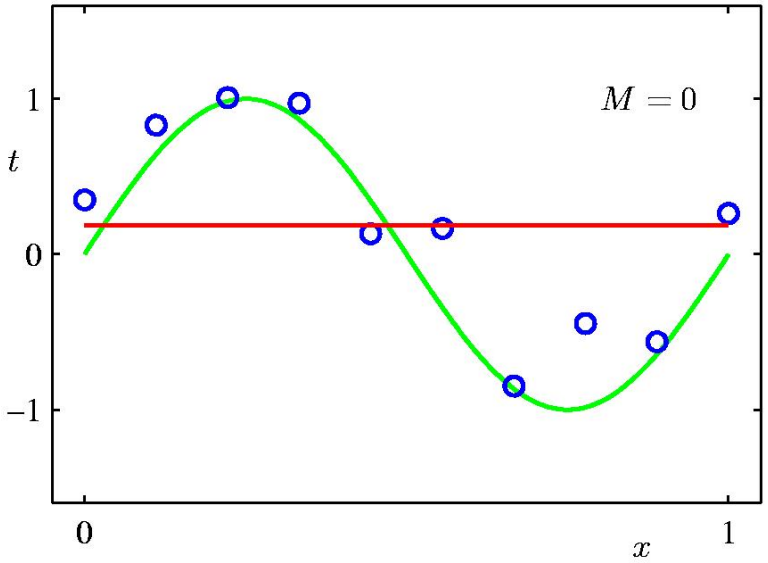

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

# Sum-of-Squares Error Function



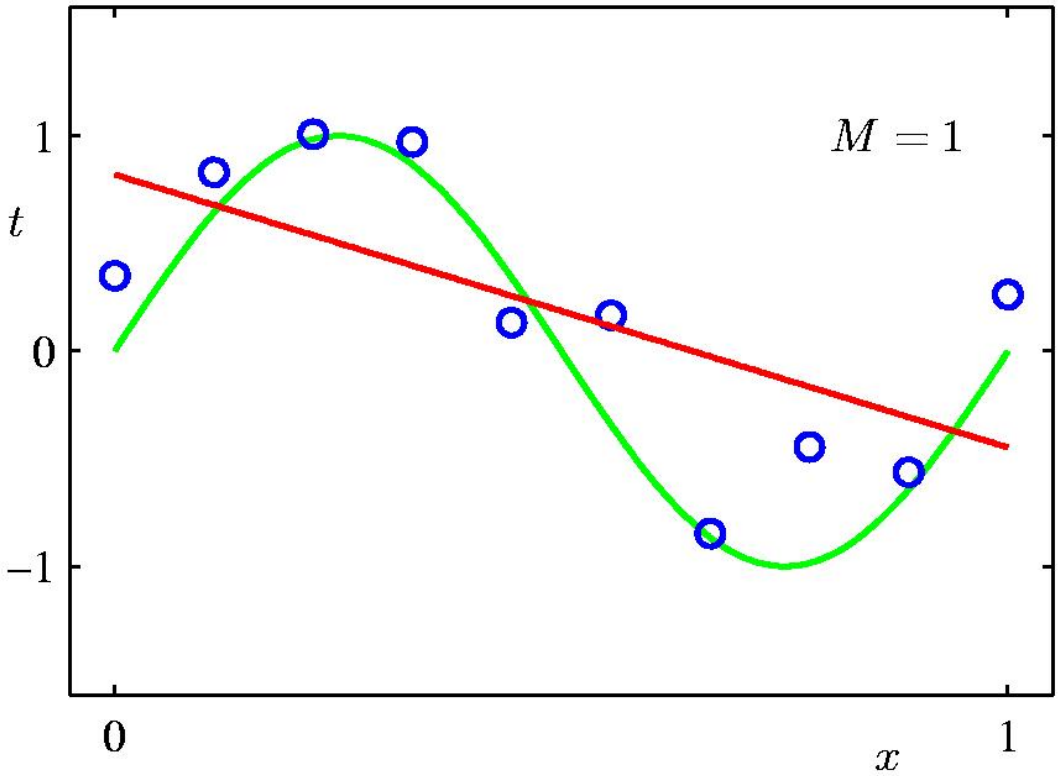

$$\sum_{i=1}^{n} [f(x_n, w) - t_n]^2$$

# 0th Order Polynomial

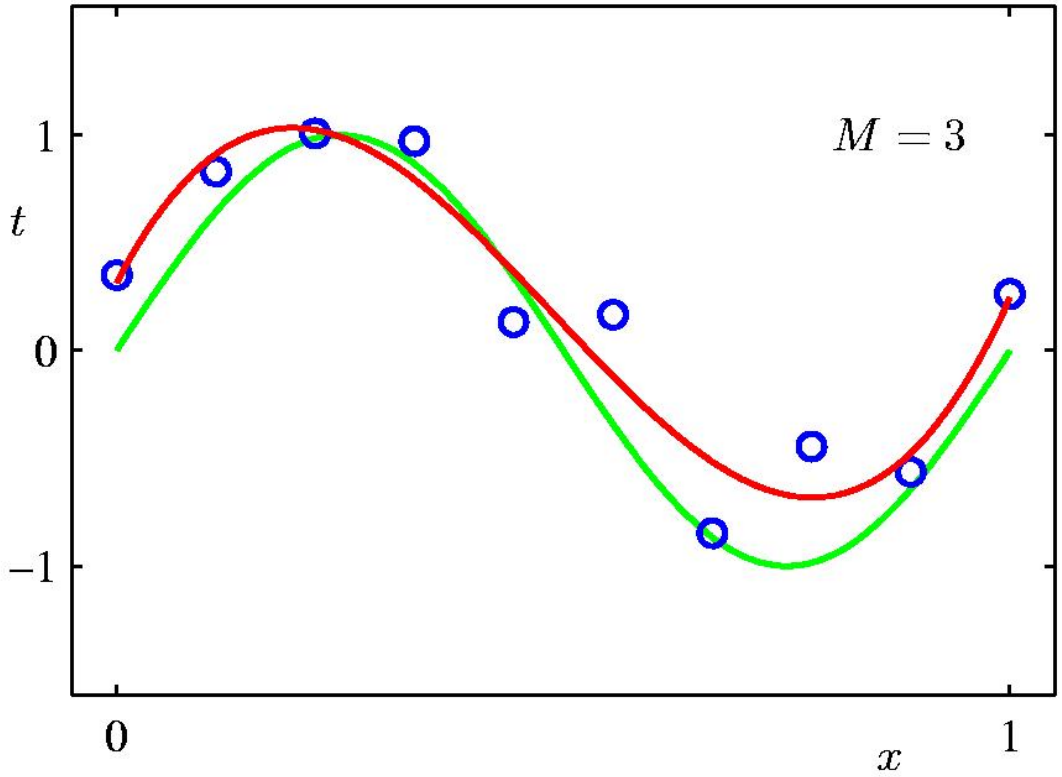



- 0th Order Polynomial fitting looks **underfitting**.
  - Underfitting occurs when a machine learning algorithm cannot capture the underlying trend of the data.

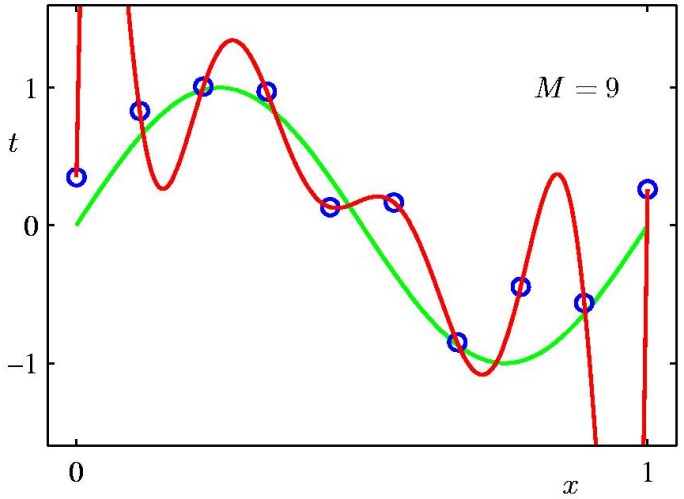
# 1st Order Polynomial



# 3rd Order Polynomial



# 9th Order Polynomial




- 9th Order Polynomial fitting looks **overfitting**.
  - Overfitting occurs when the model or the algorithm fits the data too well.
  - Generalization is important.
  - **However**, we do not know yet which is the best model, maybe the 9th degree polynomial after all!

# Incoming Lectures

- Supervised Setting (Thurs. 13.30 - 15.15)
  - Regression
  - Classification
- Unsupervised Setting (Fri. 13.30 - 15.15)
  - Clustering
  - Dimensionality Reduction
  - Deep Learning for Unsupervised Learning
- Practical Examples (Fri. 15.15 - 17.00)
  - Iris, Swiss roll, MNIST etc.

# Bonus part - Handling missing data

“There are known knowns. These are things we know that we know. There are known unknowns. That is to say, there are things that we know we don’t know. But there are also unknown unknowns. There are things we don’t know we don’t know.” - Donald Rumsfeld.

The next slides are retrieved from QuantUniversity, LLC by Sri Krishnamurthy, https://www.slideshare.net/QuantUniversity/missing-data-handling

# Missing data

| Rec No | | | | | | | Variable n |
|---|---|---|---|---|---|---|---|
| 1 | Unit non-response | | | | | | Unobserved/Latent variable |
| 2 | | | | | | | |
| 3 | | Missing data | | | | | |
| 4 | | | | Item-non-response | | | |

- Dealing with missing data, has been always a challenge in data analysis context.
- We need methods in missing data analysis that:
  - Minimize the bias
  - Maximize use of available information, and
  - Get good estimates of uncertainty e.g., p-value, confidence interval, etc.

QuantUniversity, LLC

# Methods of handling missing data

- **Deletion methods** : Delete cases or variables that are missing
  - Listwise methods
  - Pairwise deletion
  - Variable deletion
- **Imputation methods** : Substitution methods
  - Single imputation
    - Mean imputation
    - Conditional mean imputation
    - Case mean imputation
    - Regression imputation
    - Last observation carried forward
    - Worst case imputation
    - Best case imputation
    - EM imputation
  - Multiple imputation

# List wise deletion

- A good method when the proportion of missing data is less than 15%.
- Advantages:
  - It can be used for any type of statistical analysis.
  - No special computations are required.
  - The parameters estimations are unbiased.
  - The standard errors are appropriate compare to original data.
- Disadvantages:
  - May remove a considerable fraction of data

| Subject | Age | Gender | Income |
|---|---|---|---|
| 1 | 29 | M | $40,000 |
| 2 | 45 | M | $36,000 |
| 3 | 81 | M | --missing-- |
| 4 | 22 | --missing-- | $16,000 |
| 5 | 41 | M | $98,000 |
| 6 | 33 | F | $60,000 |
| 7 | 22 | F | $24,000 |
| 8 | --missing-- | F | $81,000 |
| 9 | 33 | F | $55,000 |
| 10 | 45 | F | $80,000 |

QuantUniversity, LLC

# Pair wise deletion

- Pairwise deletion involves dropping cases with missing values on an analysis-by-analysis basis
- Advantages:
  - Using all available non-missing data
- Disadvantages:
  - Estimated standard errors and test statistics are biased

| Subject | Age | Gender | Income |
|---|---|---|---|
| 1 | 29 | M | $40,000 |
| 2 | 45 | M | $36,000 |
| 3 | 81 | M | --missing-- |
| 4 | 22 | --missing-- | $16,000 |
| 5 | 41 | M | $98,000 |
| 6 | 33 | F | $60,000 |
| 7 | 22 | F | $24,000 |
| 8 | --missing-- | F | $81,000 |
| 9 | 33 | F | $55,000 |
| 10 | 45 | F | $80,000 |

# Variable deletion

- Variable deletion involves dropping variables with missing values on an case - by-case basis
- Advantages:
  - Makes sense when lot of missing values in a variable and if the variable is of relatively less importance
- Disadvantages:
  - Loss of information regarding the variable

| Subject | Age | Gender | Income |
|---|---|---|---|
| 1 | 29 | M | $40,000 |
| 2 | 45 | M | $36,000 |
| 3 | 81 | M | --missing-- |
| 4 | 22 | --missing-- | $16,000 |
| 5 | 41 | M | $98,000 |
| 6 | 33 | F | $60,000 |
| 7 | 22 | F | $24,000 |
| 8 | --missing-- | F | $81,000 |
| 9 | 33 | F | $55,000 |
| 10 | 45 | F | $80,000 |

# Mean imputation

- Replace missing values with the mean of that variable

| Case | Var1 | Var2 | Var3 |
|---|---|---|---|
| 1 | 9 | 8 | 8 |
| 2 | 7.44 | 7 | 6 |
| 3 | 8 | 5 | 6 |
| 4 | 7 | 4 | 5 |
| 5 | 9 | 5 | 7 |
| 6 | 8 | 8 | 9 |
| 7 | 6 | 7 | 6 |
| 8 | 5 | 9 | 7 |
| 9 | 7 | 8 | ? |
| 10 | 8 | 8 | 7 |

# Conditional Mean imputation

- Replace missing values with value of the variable mean for a relevant subgroup

| Case | Var1 | Sex | Var2 | Var3 |
|---|---|---|---|---|
| 1 | 9 | F | 8 | 8 |
| 2 | **8.25** | **F** | 7 | 6 |
| 3 | 8 | F | 5 | 6 |
| 4 | 7 | F | 4 | 5 |
| 5 | 9 | F | 5 | 7 |
| 6 | 8 | M | 8 | 9 |
| 7 | 6 | M | 7 | 6 |
| 8 | 5 | M | 9 | 7 |
| 9 | 7 | M | 8 | **?** |
| 10 | 8 | M | 8 | 7 |

QuantUniversity, LLC

# Case Mean imputation

- Replace missing values using information from other variables for the same case to impute the missing value

| Case | Var1 | Var2 | Var3 |
|---|---|---|---|
| 1 | 9 | 8 | 8 |
| 2 | 6.50 | 7 | 6 |
| 3 | 8 | 5 | 6 |
| 4 | 7 | 4 | 5 |
| 5 | 9 | 5 | 7 |
| 6 | 8 | 8 | 9 |
| 7 | 6 | 7 | 6 |
| 8 | 5 | 9 | 7 |
| 9 | 7 | 8 | ? |
| 10 | 8 | 8 | 7 |

# Regression imputation

- Replace missing values using information from complete cases to "predict" the value of the missing data, based on a regression equation for cases with nonmissing values

$$VAR1' = 4.621 - (.734 * VAR2) + (1.139 * VAR3)$$

| Case | Var1 | Var2 | Var3 |
|---|---|---|---|
| 1 | 9 | 8 | 8 |
| 2 | 6.32 | 7 | 6 |
| 3 | 8 | 5 | 6 |
| 4 | 7 | 4 | 5 |
| 5 | 9 | 5 | 7 |
| 6 | 8 | 8 | 9 |
| 7 | 6 | 7 | 6 |
| 8 | 5 | 9 | 7 |
| 9 | 7 | 8 | ? |
| 10 | 8 | 8 | 7 |

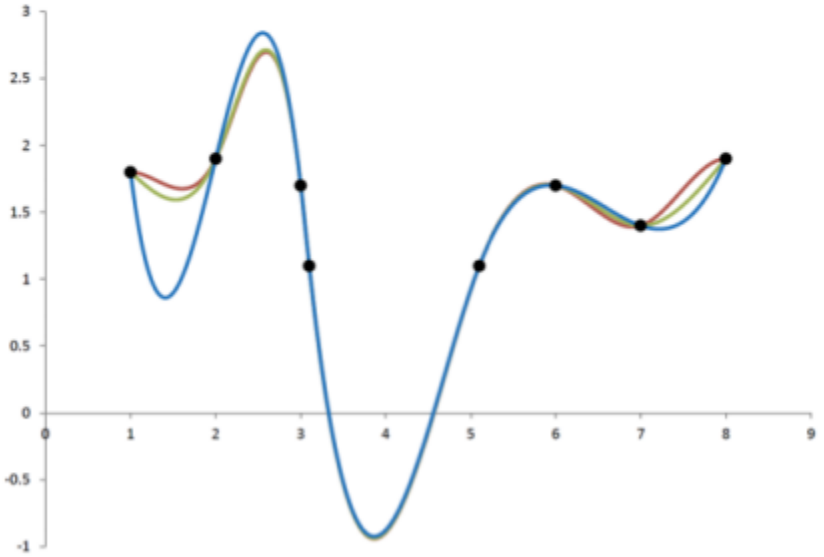
# Last observation carried forward

- Imputes the missing value as a value on the same outcome the most recent time it was observed
- Variants :
- Average of T1 and T2

| Case | T1 | T2 | T3 |
|---|---|---|---|
| 1 | 9 | 8 | 8 |
| 2 | ? | 7 | 6 |
| 3 | 8 | 5 | 6 |
| 4 | 7 | 4 | 5 |
| 5 | 9 | 5 | 7 |
| 6 | 8 | 8 | 9 |
| 7 | 6 | 7 | 6 |
| 8 | 5 | 9 | 7 |
| 9 | 7 | 8 | 8 |
| 10 | 8 | 8 | 7 |

# Interpolation

- Use interpolation to fill in missing values
- Useful for longitudinal datasets




QuantUniversity, LLC

# Worst case and Best case imputation

- Worst case replaces a missing value with the worst case scenario for a categorical outcome
- Best case replaces a missing value with the best case scenario for a categorical outcome

# Expectation-Maximization

- Substitute best missing values using a ML imputation
- In the E-step, expected values are calculated based on all complete data points
- In the M-step, the procedure imputes the expected values from the E-step and then maximizes the likelihood function to obtain new parameter estimates

# Multiple imputation

- Multiple imputation is quickly becoming the “gold standard” approach to handling missing values
- Computationally complex

# References


- Eric Eaton, 2017, https://www.seas.upenn.edu/~cis519/fall2017/lectures/01_introduction.pdf
- Iris flower data set, Wikipedia, https://en.wikipedia.org/wiki/Iris_flower_data_set
- Kurama, Vihar 2017. Introduction to Machine Learning, retrieved from: https://towardsdatascience.com/introduction-to-machine-learning-db7c668822c4
- MNIST database, Wikipedia, https://en.wikipedia.org/wiki/MNIST_database
- Slides are expanded from Berrin Yanikoglu, Sabanci University retrieved from http://people.sabanciuniv.edu/berrin/cs512/lectures/
- Sri Krishnamurthy, https://www.slideshare.net/QuantUniversity/missing-data-handling