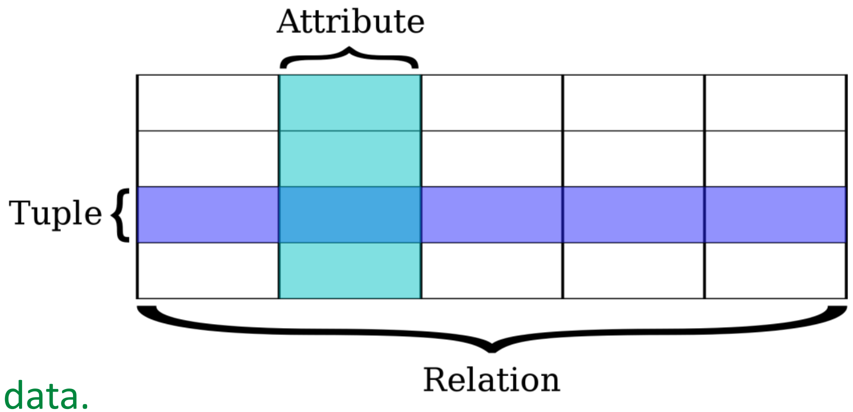
Supervised Learning

Modelling

~~Un~~Supervised Learning



- Unsupervised Learning
  - Given X
  - … the task of inferring a function to describe hidden structure from unlabeled data.
  - Distribution / Density, Summary statistics, Clustering, Association Rules, Dimensionality Reduction

- Supervised Learning
  - Given X & y (a **particular** random variable)
  - Find what is the relation between the particular random variable and other random variables
    - What if we are only interested in identifying customers who bought Milk?
  - Find how the value of the dependent variable depends on the value of others
  - Find how the outcome is related to the features
  - Key Variations: Type of outcome / dependent r.v.
    - Numeric (Discrete, Continuous, [0,1])
    - Categorical : Nominal, Ordinal

The idea of a Model

- Physical
  - a physical copy of an object such as a globe
- Computer
  - a simulation to reproduce behavior of a system
- Scientific
  - a simplified & idealized understanding of physical systems
  - Newton's Law model the physical universe
- Conceptual
  - a representation of a system using general rules & concepts
- Mathematical
  - a representation of a system using mathematical concepts
- Statistical
  - a parameterized set of probability distributions

$$y = 3x + 4$$
$$y = x^2$$
$$y = e^x$$
$$y = \log(x)$$
$$y = \sin(x)$$

All models are false. Some models are useful.

The idea of a Statistical / ML Model

- Model
  - A function relates two (or more) variables
  - Captures the relation between x and y
  - For every value of x, there must be a unique value of y
  - Data looks like $\{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots (x_n, y_n)\}$

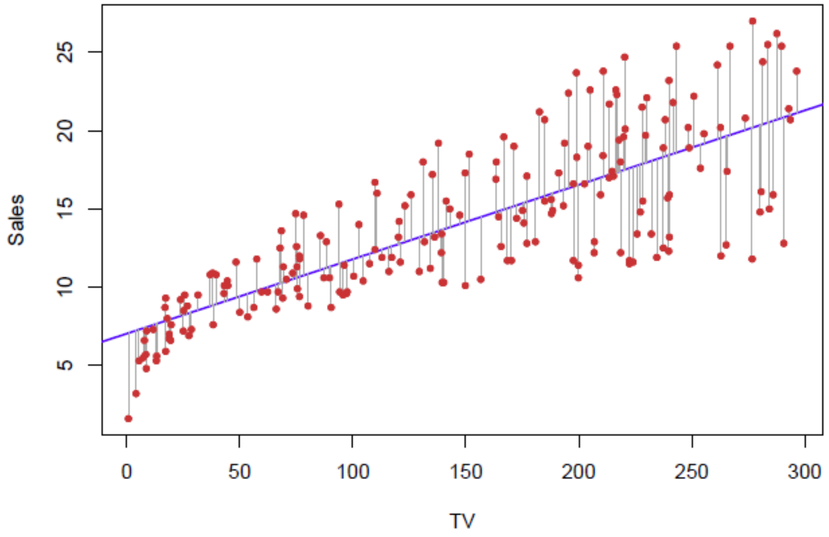
$$y = 3x + 4$$
$$y = x^2$$
$$y = e^x$$
$$y = \log(x)$$
$$y = \sin(x)$$

$$y = f(x)$$

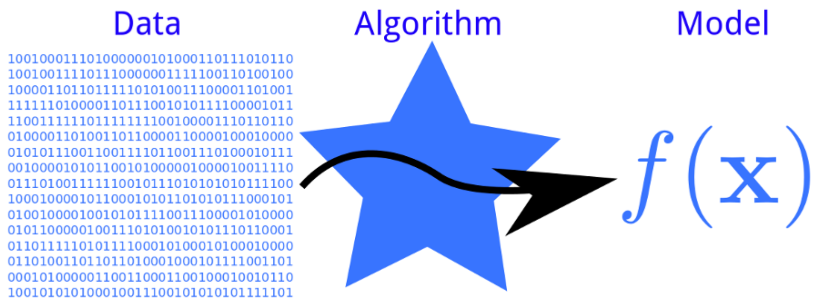
- Statistical Model
  - Real world data looks like $\{(x_1, y_1), (x_1, y_2), \dots, (x_n, y_n)\}$
  - Multiple values of y for a single value of x
  - In expectation (on average), “model” captures the relationship between variables
  - Effects due to unobserved variables / Errors in measurements : capture by ε
  - Randomness / Stochasticity / Noise : Zero-mean; Normal distribution
  - Violations of Assumption is an indication of systemic errors

$$y = f(x) + \varepsilon$$
$$\varepsilon \sim \mathrm{N}(0, \sigma)$$

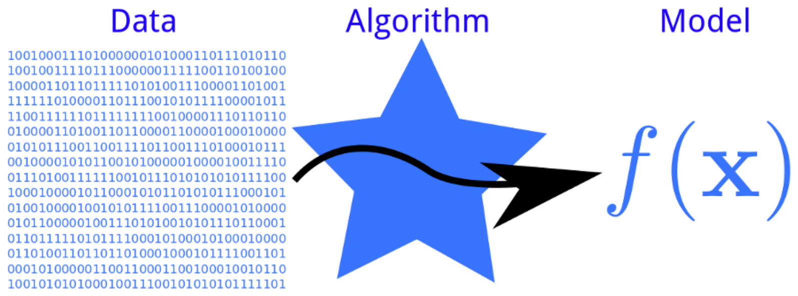
$$\hat{y} = \hat{f}(x) + 0$$
$$\mathrm{P}(y \mid x)$$



Un/Supervised **Learning**

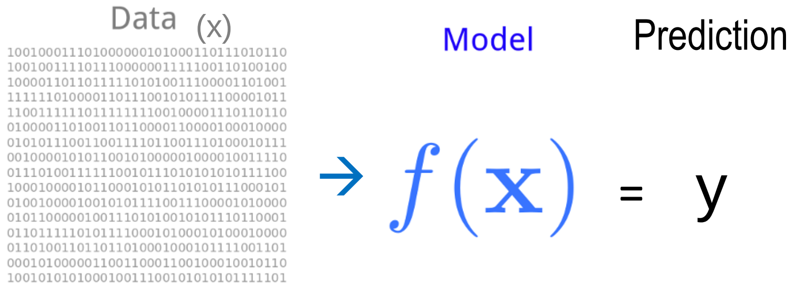


$$y = f(x)$$

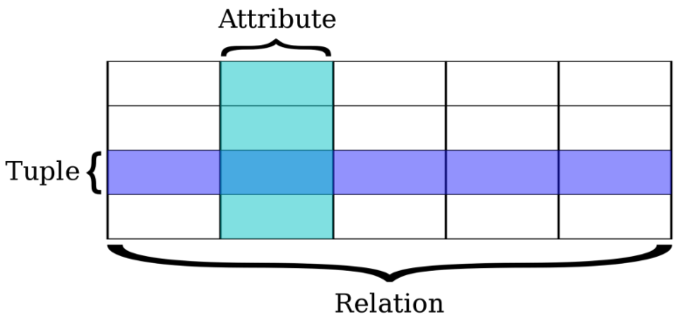


- Given X
  - … the task of inferring a function to describe hidden structure from unlabeled data.
  - Distribution / Density, Summary statistics, Clustering, Association Rules, Dimensionality Reduction

- Given X & y (a **particular** random variable)
  - Find what is the **relation** between the particular random variable and other random variables
  - Find how the value of the **dependent (particular)** variable depends on the value of others
  - Find how the outcome is related to the **features**
  - Generalize : Make **predictions** about new data



Un/Supervised Learning **Models**



- Supervised
  - Dependent vs. Independent Variables
  - Is there a variable of interest? Labelled data?
  - Do you know what you are looking for?
  - View the data as $\{(x_1, y_1), (x_1, y_2), \ldots, (x_n, y_n)\}$
  - Regression vs. Classification

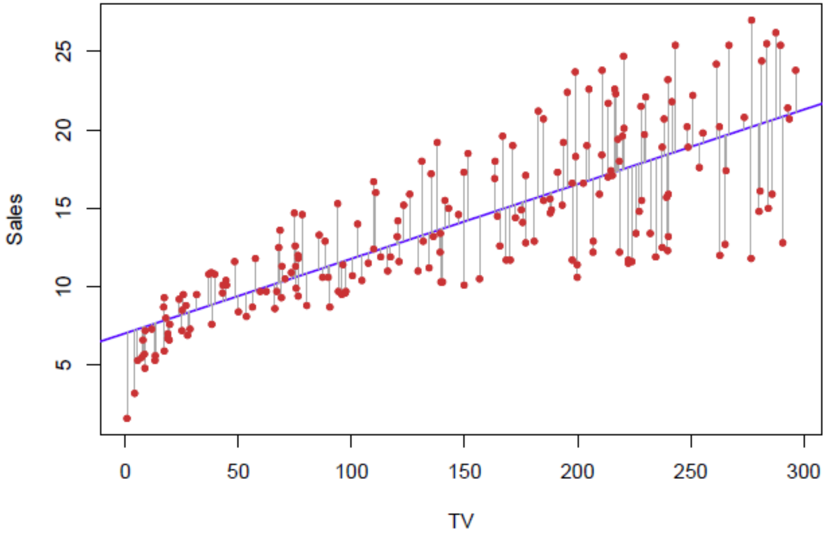
- Unsupervised
  - No clearly defined Dependent Variable
  - Find patterns in data
  - View the data as $\{(x_1), (x_2), \ldots, (x_n)\}$
  - Often, a pre-processing step to Supervised

- Parameteric
  - Specify the "form" of f *(Specify model class)*
  - Learn exact f *(Learn model parameters)*
  - Restrictive but Interpretive
  - Less data required for learning

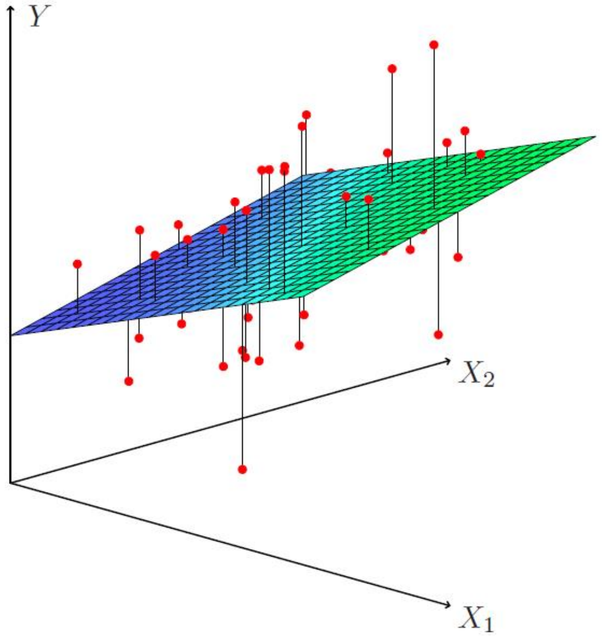
- Non-Parameteric
  - Learn model directly *(No restrictions on model class)*
  - Flexible but less Interpretive

- Model-Based vs. Model-Free
  - Models are not the only game in town
  - Model-Based: Linear Regression (What is the model?)
  - Model-Free: Nearest Neighbor, Collaborative Filtering

Supervised Learning : Linear Regression



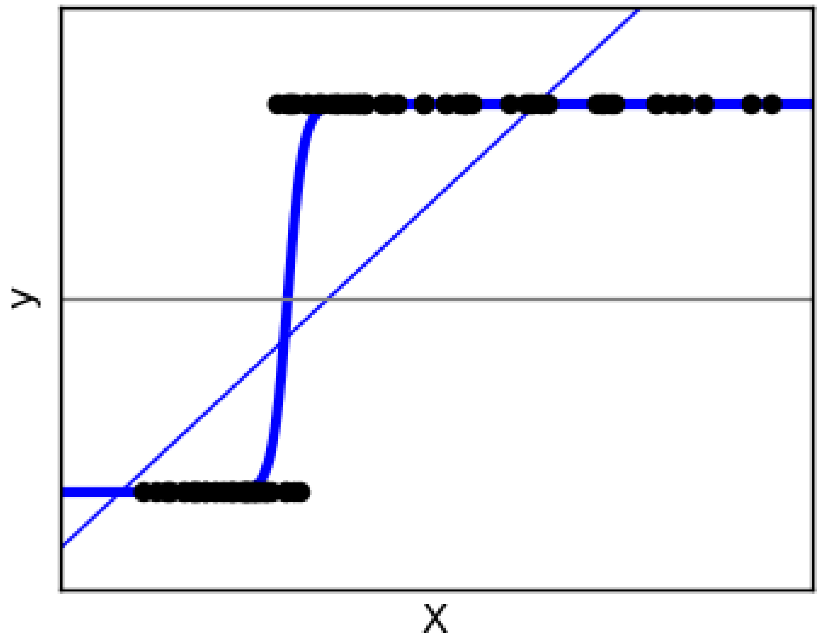
p=1



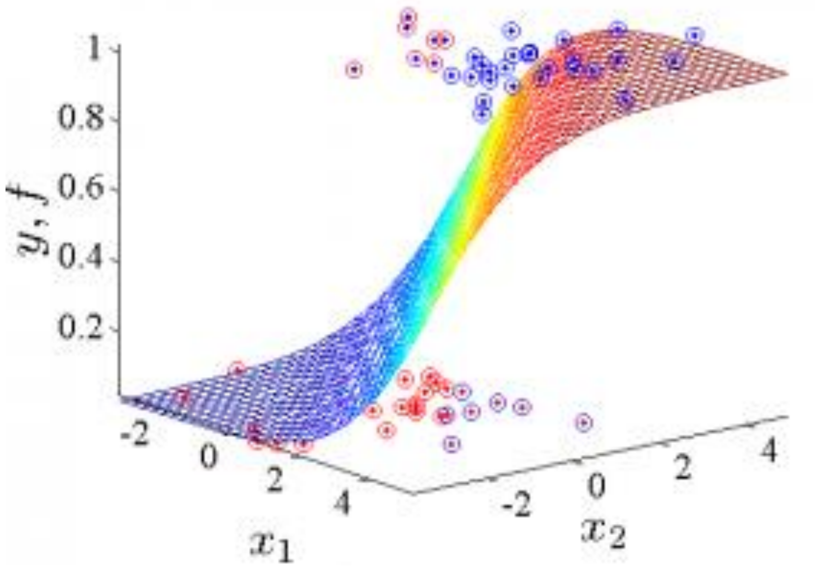
p=2

p > 2 ?

Supervised Learning : Binary classification



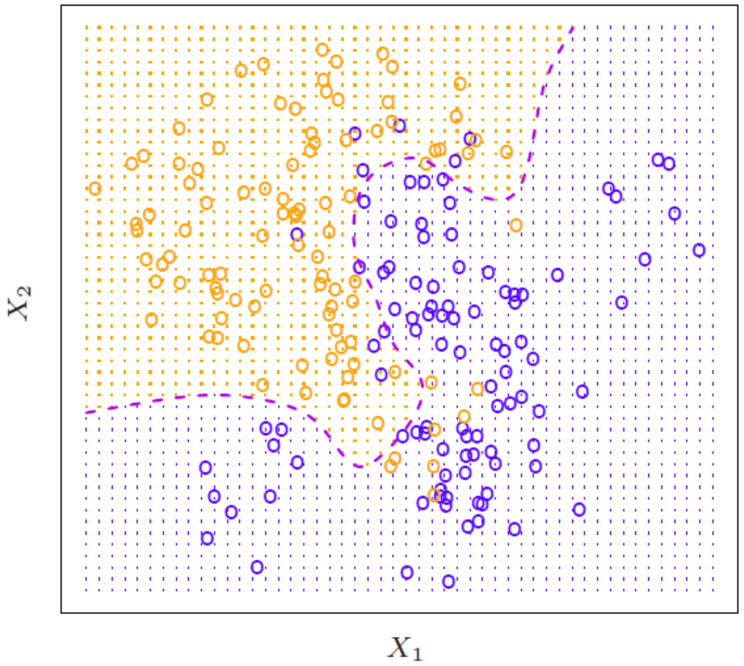
p=1



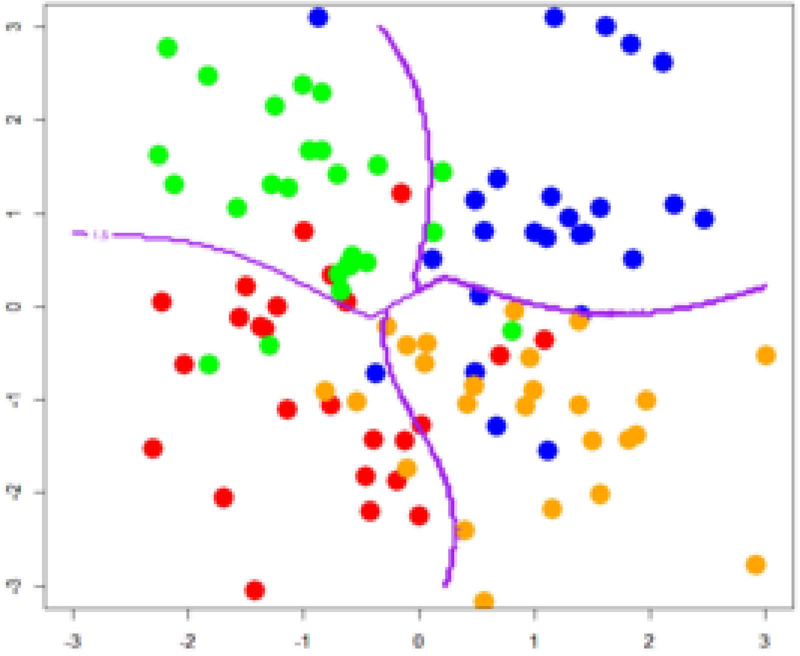
p=2

p > 2

Supervised Learning : From Binary to Multi Class



p=2

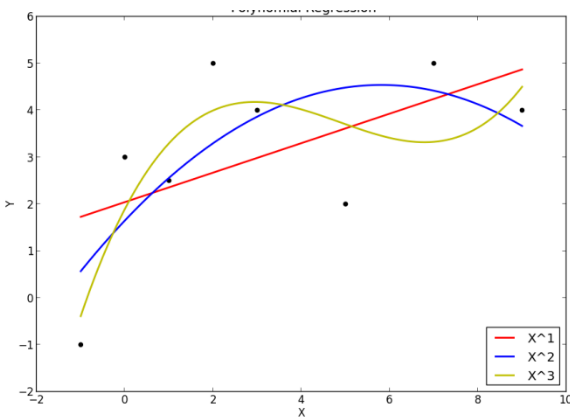


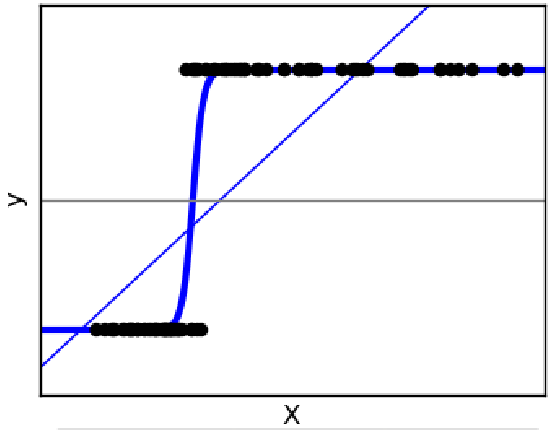
p=2

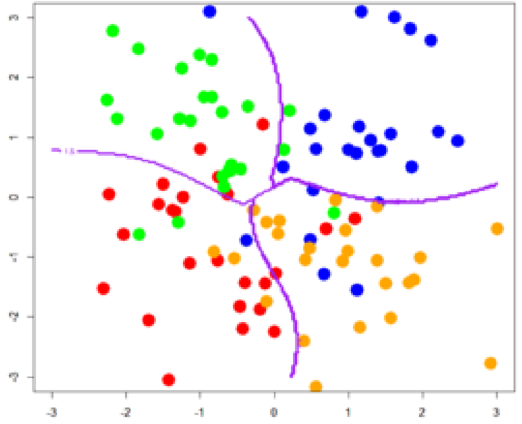
p > 2 ?

SL: Variant Summary

- Numeric y
  - Given input data x, f(x) is a numeric value
  - Regression: Linear, polynomial, lasso
  - Time Series : y = xt+1
- Numeric y in [0,1]
  - Given input data x, f(x) is a numeric value in between 0,1 (e.g. probability)
  - Regression: Logistic
- Categorical y
  - Given input data x, f(x) is a label / class / category (e.g. churn or not)
  - Classification: knn, logistic, decision tree, svm
- Ordinal y
  - Learn f(x) such that given input data x, f(x) is a rank (e.g. 1st, 2nd , …)
  - Ranking







Let’s play

Thought Experiments

Data	**Past credit card transactions of customers**
Business Objective	Identify fraudulent transactions
Analytics	?

Data	**Past purchases of customers**
Business Objective	What is a customer likely to buy next?
Analytics	?

Data	**Pricing and Sales data of a product portfolio**
Business Objective	Determine price elasticity
Analytics	?

Thought Experiments (cont'd)

Data	**?**
Business Objective	How much should company spend on TV/radio/paper ads?
Analytics	?

Data	**Past purchases of customers**
Business Objective	Segment customers with similar purchase behavior
Analytics	?

Data	**A set of emails marked junk or not by a human**
Business Objective	Build a rules engine to determine emails as Junk or not?
Analytics	?

Statistical Decision Theory

Praphul Chandra

Statistical Decision Theory

- Framework
  - Function Approximation
  - Joint Probability Distribution
  - Loss Function

Function Approximation: $Y = f(X)$
Joint Distribution: $\mathbb{P}(X, Y)$
Loss Function: $L(Y, f(X))$

- Loss Variants
  - L2 (Squared Error Loss)
  - L1 Loss

$$L(Y, f(X)) = (Y - f(X))^2$$

$$\begin{aligned} EPE(f) &= \mathbb{E}[(Y - f(X))^2] = \int [y - f(x)]^2 \mathbb{P}(dx, dy) \\ &= \mathbb{E}_X \mathbb{E}_{Y|X}[(Y - f(X))^2 | X] \end{aligned}$$

- Expected Prediction Error
  - Choosing the “best” function
  - Depends on choice of loss function
  - **L2**: The best prediction of Y at an point X=x is the conditional **mean**.
  - **L1**: The best prediction of Y at an point X=x is the conditional **median**

$$\begin{aligned} f(x) &= \arg\min_c \mathbb{E}_{Y|X}[(Y - c^2 | X = x] \\ &= \mathbb{E}[Y | X = x] \end{aligned}$$

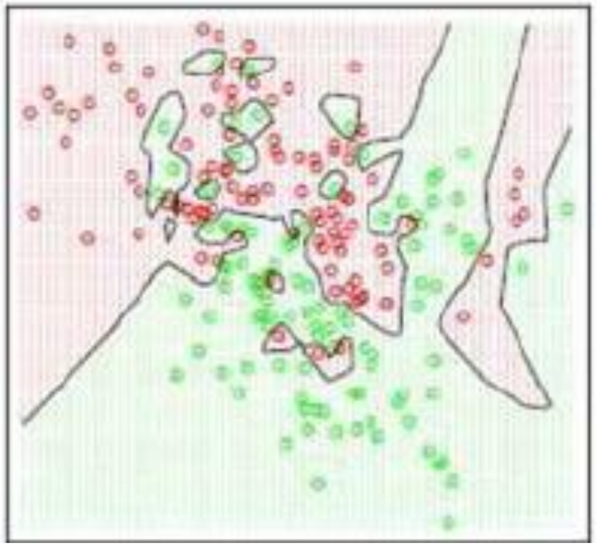
K-Nearest Neighbor

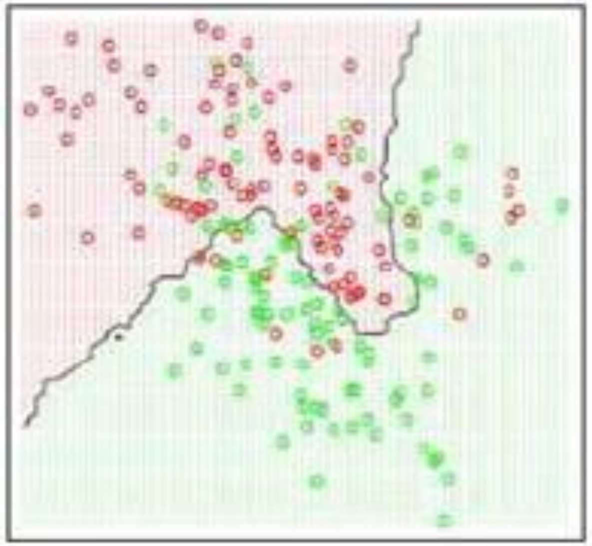
- Statistical Decision Theory
  - The best prediction of Y at an point X=x is the conditional mean. (L2 loss)
  - knn: At each point x, approximate y by averaging all y_i with input x_i near x

$$f(x) = \mathbb{E}[Y|X = x]$$

$$\hat{f}(x) = Ave(y_i|x_i \in N_k(x))$$

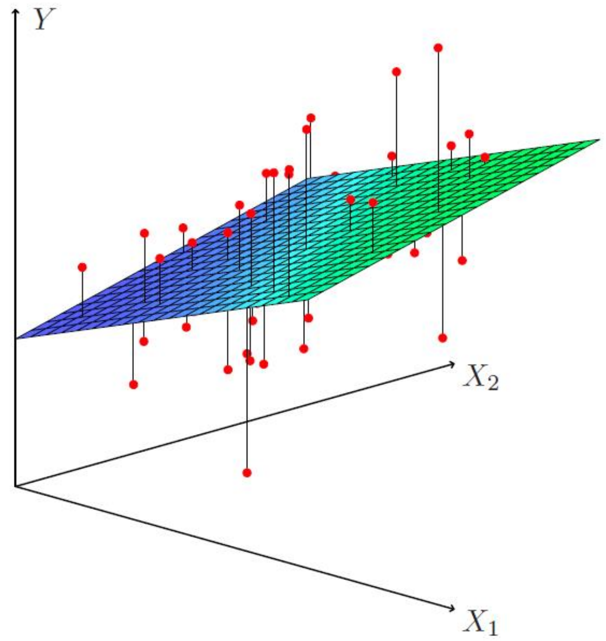
- Two approximations
  - Expectation is approximated by averaging over sample data.
  - Conditioning at a point x is relaxed to conditioning on some region “close” to x
- Note
  - Regression (Mean); Classification (Majority Vote)
  - Model Free; Lazy *(Separating boundary not really created)*
  - Locally constant
  - Computational Complexity *(Time, Space)*
- Behavior
  - Large k : Smoother boundaries (class separating)
  - Large N : Large storage req. (space complexity)
  - Large p : lower accuracy (curse of dimensionality)





Linear Regression

- Statistical Decision Theory
  - The best prediction of Y at an point X=x is the conditional mean. (L2 loss)
  - LR : Find a linear function which minimizes the total loss (sum of least squares) across x
- Two approximations
  - Global function
  - Linearity
- Note
  - Model Based *(f() is Globally Linear)*
  - Computational Complexity *(Time, Space)*
- Behavior
  - Large N : Larger training time (computational complexity)
  - Large p : potentially lower accuracy (linearity in higher dimensions)
  - Larger k?? *(Feature Expansion – Later)*



Statistical Decision Theory: Summary $Y = f(X)$

f	(X)	$L(Y, f(X))$

• Constant
• Linear
• Non-Linear
    • Polynomial
• Piecewise
    • Splines & Kinks
• Additive
	• Global
	• Local
	• Kernel
	• Basis Transformation
	    • Expansion
	    • Reduction
	    • Learn (Dictionary)
	• Manifold
		• Distance Measure
		    • L2, L1, etc.
		    • Hinge Loss
		• Overfitting
		    • Regularization
		    • Penalize roughness

Knn vs. Linear Regression : Two ends of the spectrum

- Lazy
  - Nothing is done at training time
  - Training data used to predict (Memory intensive)
- Model free
  - No parameters!
  - No boundary *(classification)* / No coefficients *(regression)*
  - No optimization !
- Low Bias (Very flexible)
  - But High Variance
- Hyperparameter Optimization
  - Optimal k

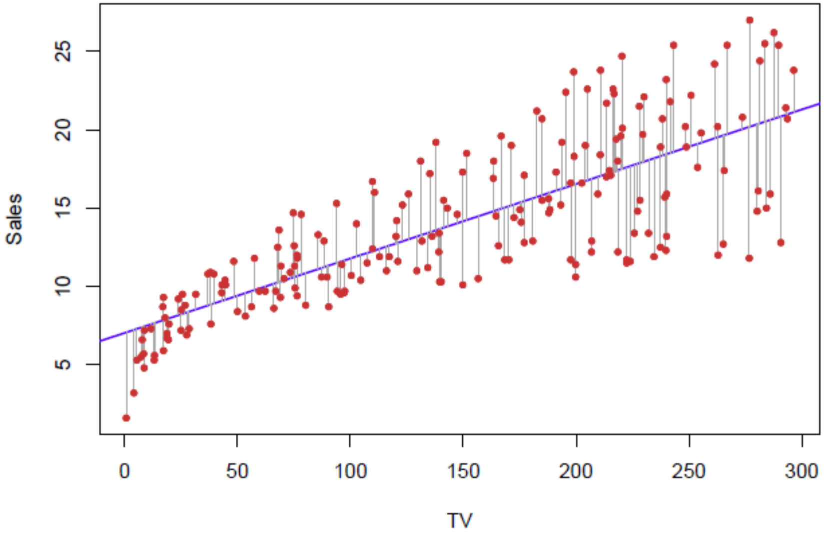
- Eager
  - Training data is "processed" before new data arrives
  - Model (not train data) used to predict
- Model Based
  - Parameters : Coefficients
  - Coefficients ➔ linear combination ➔ Hyperplane (model)
  - Optimization solution via normal equations
- High Bias
  - Low Variance
- Hyperparameter Optimization
  - Optimal degree of the coefficients

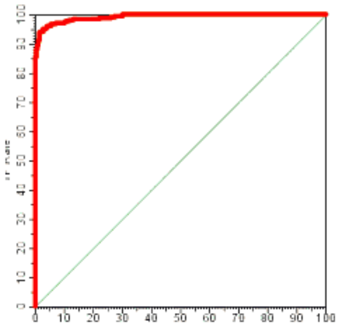
Supervised Learning

Model Evaluation

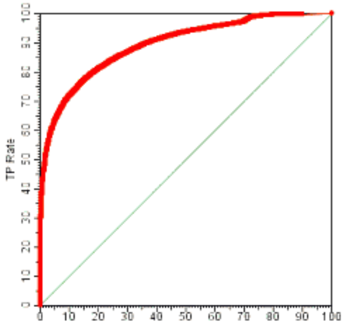
What is a good model?

- Regression
  - Low “Error” : Define.
  - How well does the model “explain” the data?
    - Quality of fit
  - Residuals
    - Error between actual and predicted
  - Residual Sum of Squares (RSS)
    - Measure of total error
  - Mean Square Error (MSE)
    - Measure of total error normalized by number of observations

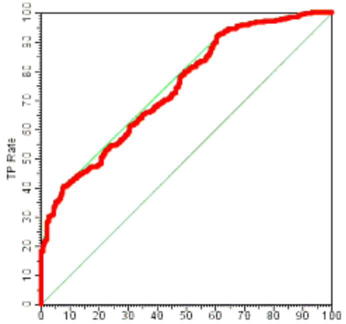




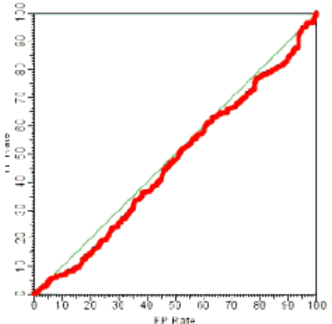
good separation



reasonable



poor separation

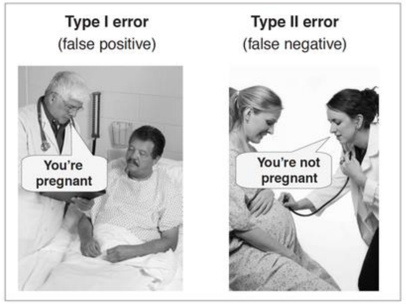


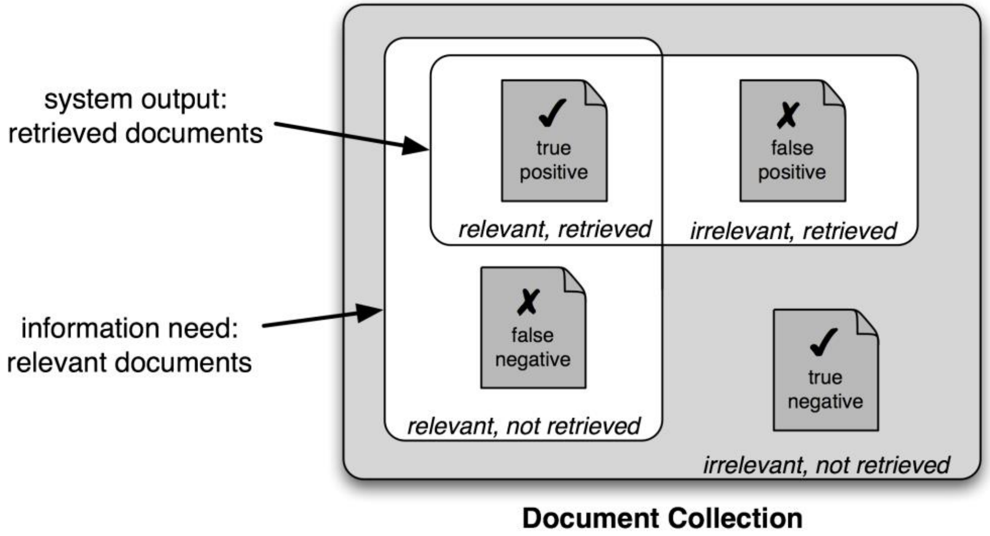
random separation

$$\text{Errors} = \sum_{i=1}^{n} \mathrm{I}(y_i \neq \hat{y}_i)$$

$$\text{Error Rate} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{I}(y_i \neq \hat{y}_i)$$

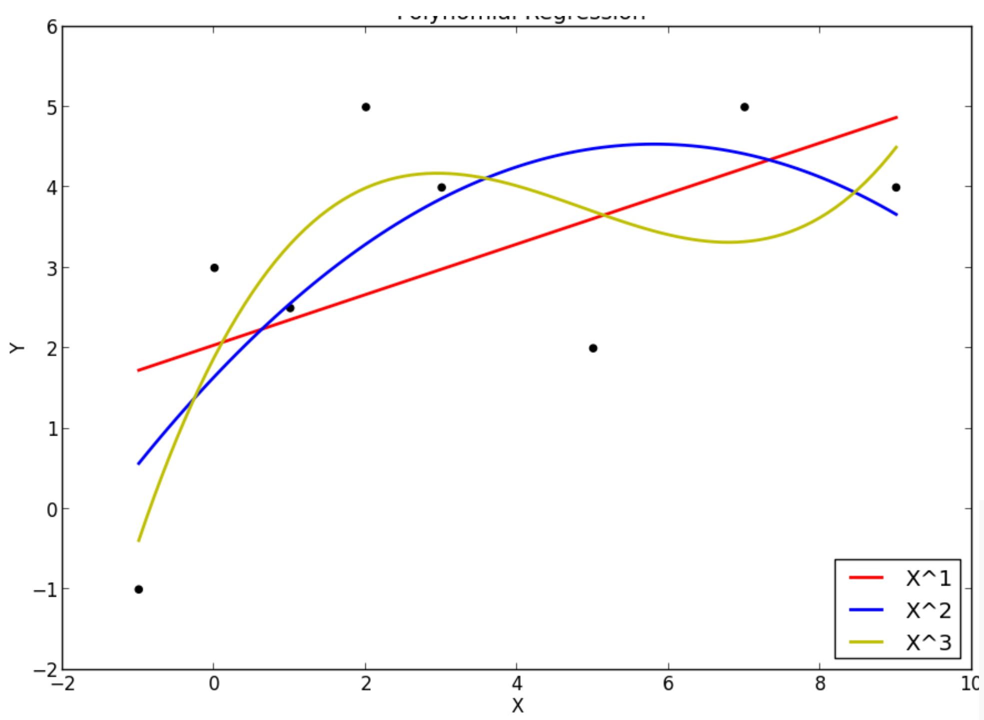
- Classification
  - Binary classes : A/B, 1/2, +/-
  - \+ predicted as – (Type-1 error)
  - \- predicted as + (Type-2 error)
  - Confusion Matrix
  - Precision, Recall, Accuracy, Sensitivity

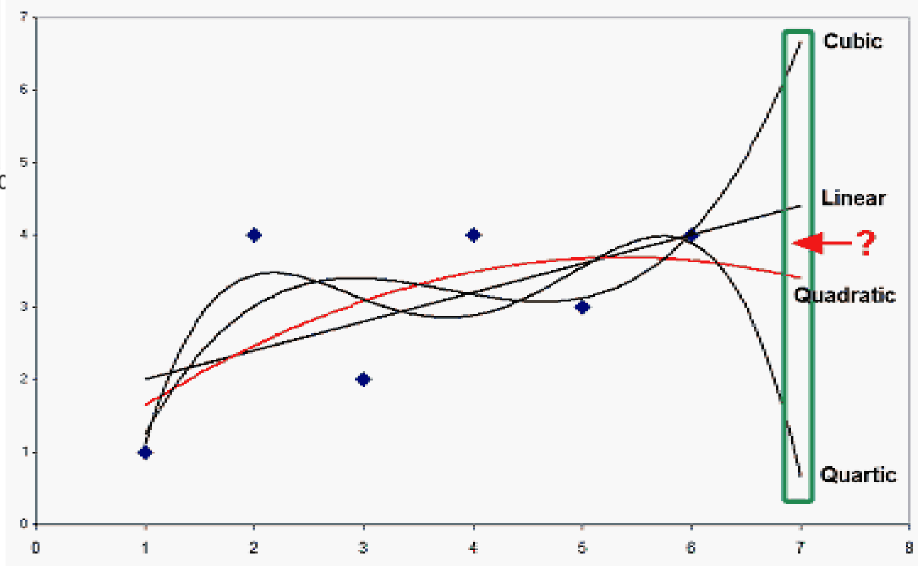




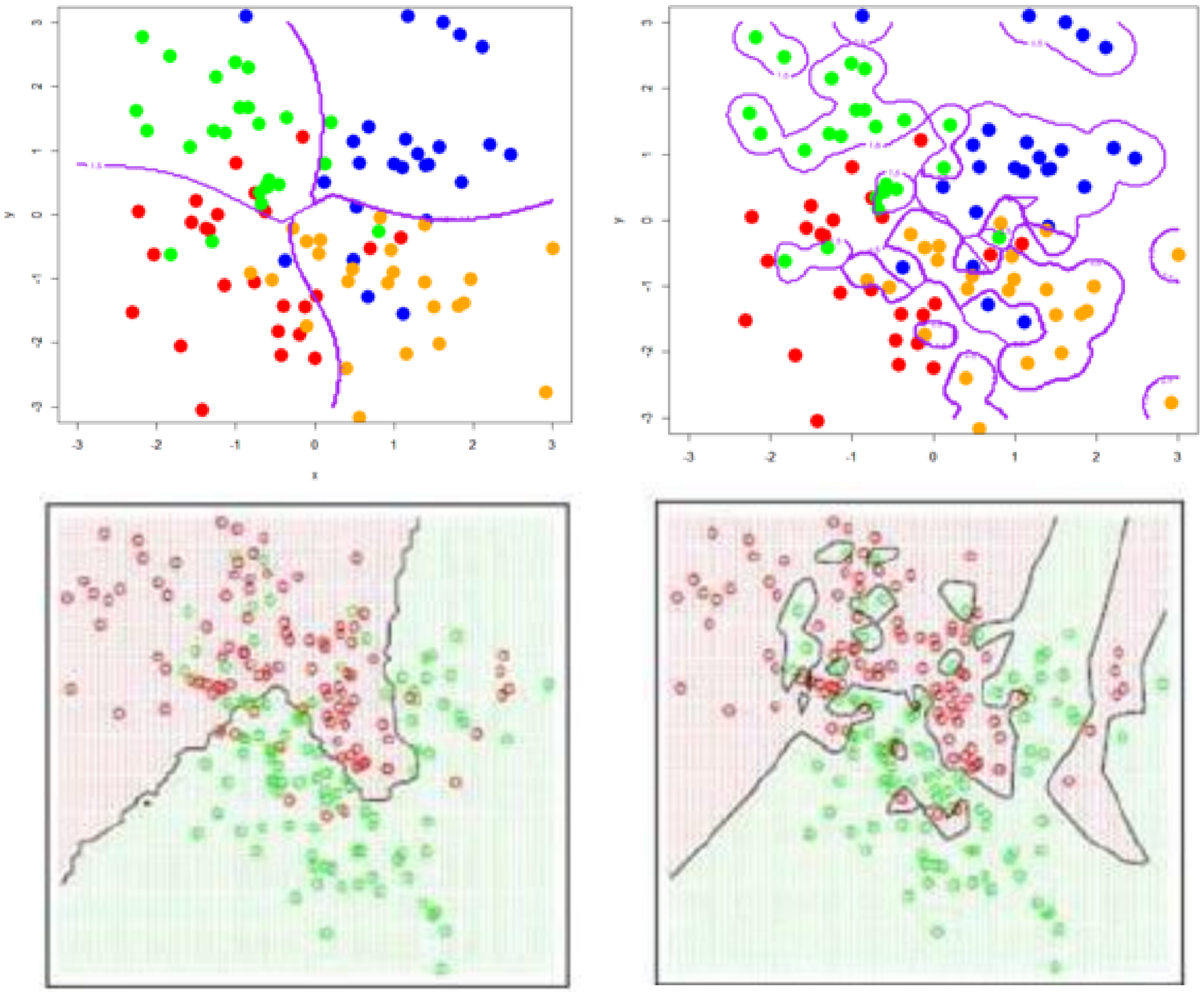
Measure	Formula
Predictive accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Error rate (1- predictive accuracy)	$\frac{FP + FN}{TP + TN + FP + FN}$
Sensitivity, true positive rate, recall	$\frac{TP}{TP + FN}$
Specificity, true negative rate	$\frac{TN}{FP + TN}$
Precision, PPV	$\frac{TP}{TP + FP}$
NPV	$\frac{TN}{TN + FN}$

Reducing error… at what cost? | Regression





Reducing error… at what cost? | Classification

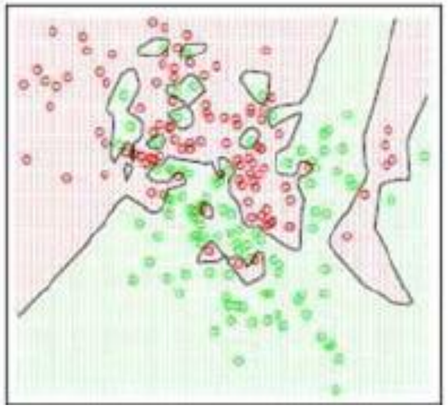


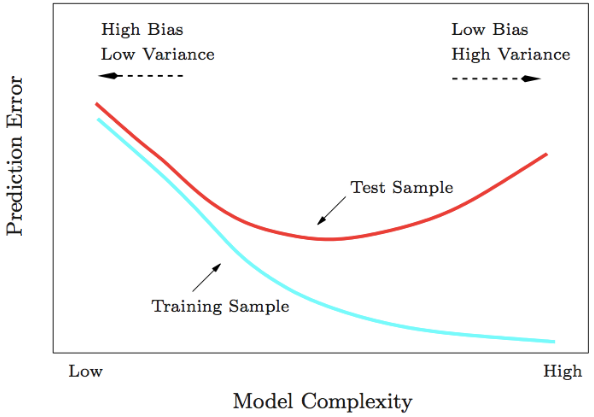
Model Evaluation : Error vs. Complexity

- Intuition
  - Some models are “un-necessarily complex”
  - Some models tend to “over fit” the given data
  - Does a model “overfit”?
    - Visual inspection not always feasible
    - High dimensional data *(too many variables, features)*
- Approch-1: Constrain the complexity of the model
  - Define statistic on the data *(statistical approach)*
  - Adjusted R2 : Explained Variance normalized with DoF
  - AIC / BIC / Cp : penalizes number of parameters in model
- Approach-2: Measure model performance on “new” data
  - Split available data
    - Learn model using “Training data; Evaluate on “Test data”
  - Train vs. Test Data : Train vs. Test Error
  - Try it out on test data *(computational approach)*

- BIG Idea: Generalization Error
  - How does model perform on data it did not learn from?
  - Model Complexity / Flexibility vs. Model Performance
  - Lower Training error does not always imply Lower Test Error!

- Equivalence
  1. Model Overfits
  2. Model reduces training error with an over-complex model
  3. Model reduces training error but test error increases





Complexity-aware Model Evaluation

Validation Set

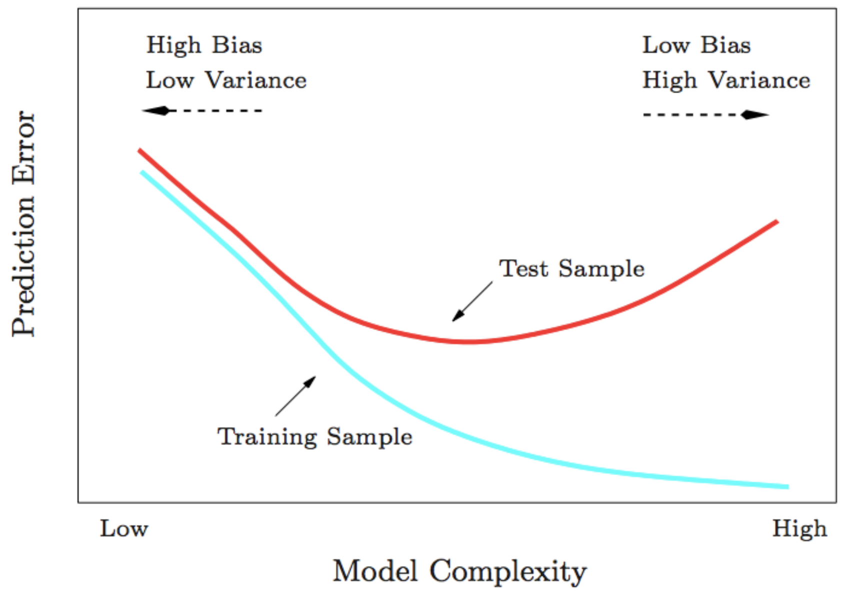
- Key Idea : Assume you have less data available than you actually have
- Split your data into training & test (validation)
- Learn the model on training set. Evaluate (Test) it on validation

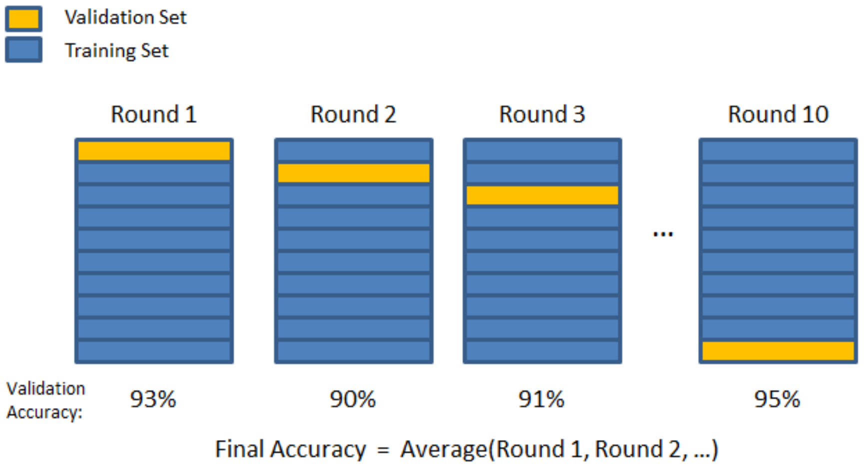
LOOCV

- Validation Set = 1 instance
- Learn the model on training set. Evaluate (Test) it on validation
- Repeat (Go to step-1)

K-Fold CV

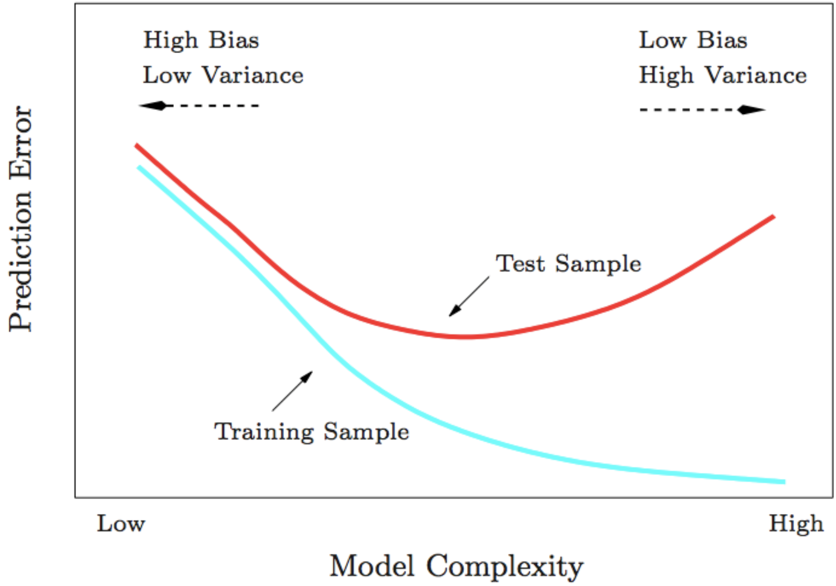
- Validation Set = 1 sub-set
- Learn the model on training set. Evaluate (Test) it on validation
- Repeat (Go to step-1)
- Gold Standard :
  - More stable than validation set;
  - Less computationally intensive than LOOCV





What is a good model : Summary

- Model Complexity & Overfitting
  - Trying to reduce training error with a more complex model
  - More degrees of freedom (More variables, features)
  - Error can be reduced with more complex models: When is it overfiitting?
  - Lower Training error does not always imply Lower Test Error!



- Bias Variance Tradeoff
  - Bias: Error introduced due to simplifying the real world with a “simple” model.
  - Variance: How much does the model vary if we train it on a different training set?
  - Tradeoff: Increasing Complexity ➔ Lower Bias but may lead to overfitting (higher variance)

- Approaches for model evaluation
  - Validation Set, LOOCV, K-fold
  - Given Data = Training + Test
  - Given Data = Training + Calibration + Test (Later)

Statistical Decision Theory: Summary $Y = f(X)$

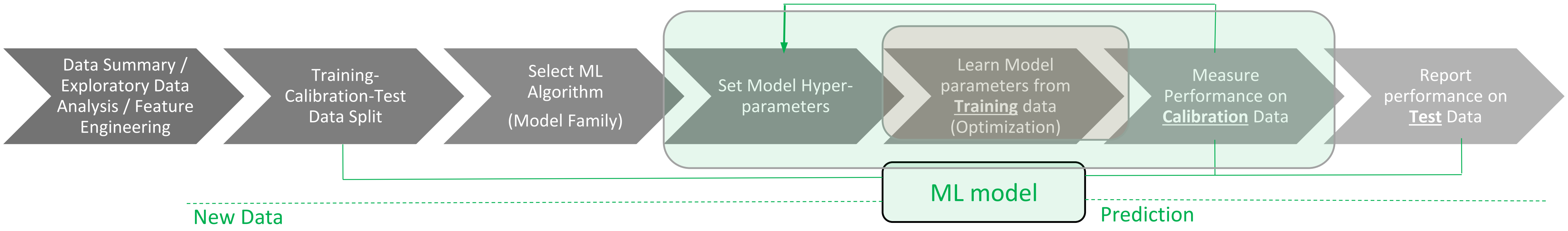
f	(X)	$L(Y, f(X))$

• Constant
• Linear
• Non-Linear
  • Polynomial
• Piecewise
  • Splines & Kinks
• Additive
	• Global
	• Local
	• Kernel
	• Basis Transformation
	  • Expansion
	  • Reduction
	  • Learn (Dictionary)
	• Manifold
		• Distance Measure
		  • L2, L1, etc.
		  • Hinge Loss
		• Overfitting
		  • Regularization
		  • Penalize roughness

Machine Learning Framework

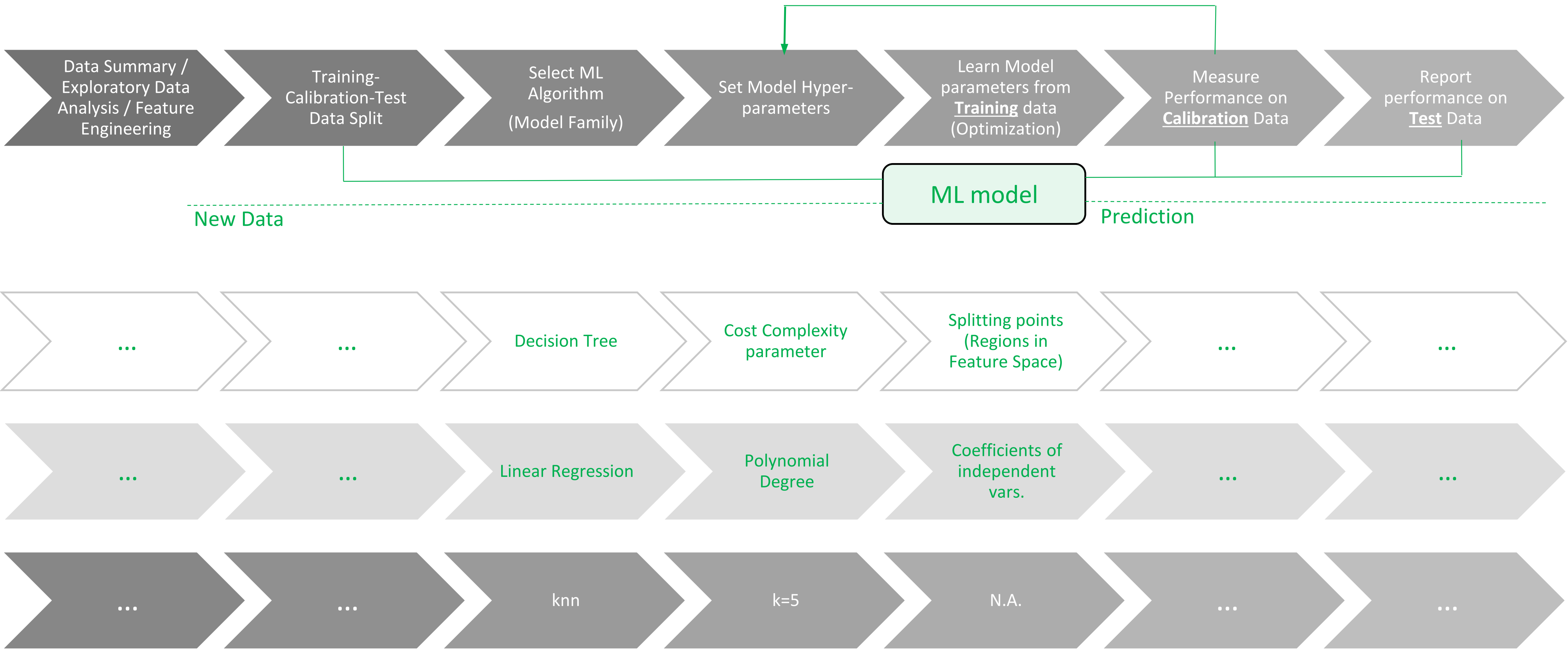
Praphul Chandra

Machine Learning Framework



- Many models let user choose model complexity
  - knn: Lower k ➔ Higher model complexity
  - DT: Lower α ➔ Higher model complexity
  - LR: ?
- Hyperparameter Optimzation
  - Optimal model complexity
  - Iterate over Hyperparameters + CV (grid search)
- Parameter Optimization
  - Minimize the Loss Function : L(Y, f(X))
  - Given the model (hyperparameter)
  - Not for every model family (knn)

Machine Learning Framework



Machine Learning Framework (cont’d)