Model Evaluation

Supervised Learning

Evaluating a Regression Model

- Residuals
  - Error between actual and predicted
- Residual Sum of Squares (RSS)
  - Measure of total error
  - R-sq.
    - Normalized by Total Sum of Squares
    - Unitless
  - Mean Square Error (MSE)
    - Normalized by number of observations
    - Squared Units of dependent variable
  - Root Mean Square Error (RMSE)
    - Normalized by number of observations
    - Units of dependent variable
- Mean Absolute Error (MAE)
  - Normalized by number of observations
  - Units of dependent variable
- Mean Absolute Percentage Error (MAPE)
  - Normalized by number of observations
  - Unitless
- Inferential Statistics (t, p) + Validate assumptions

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

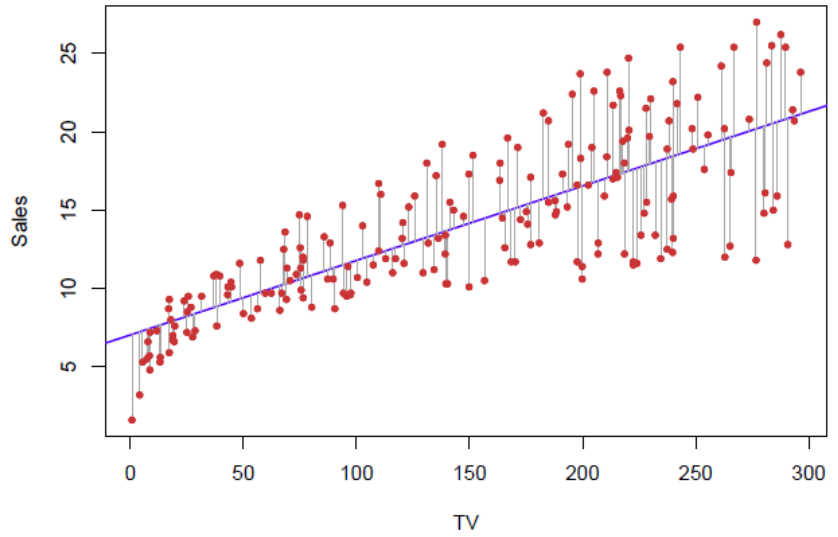
$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

$$MSE = \frac{RSS}{n} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{RSS}{n}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$



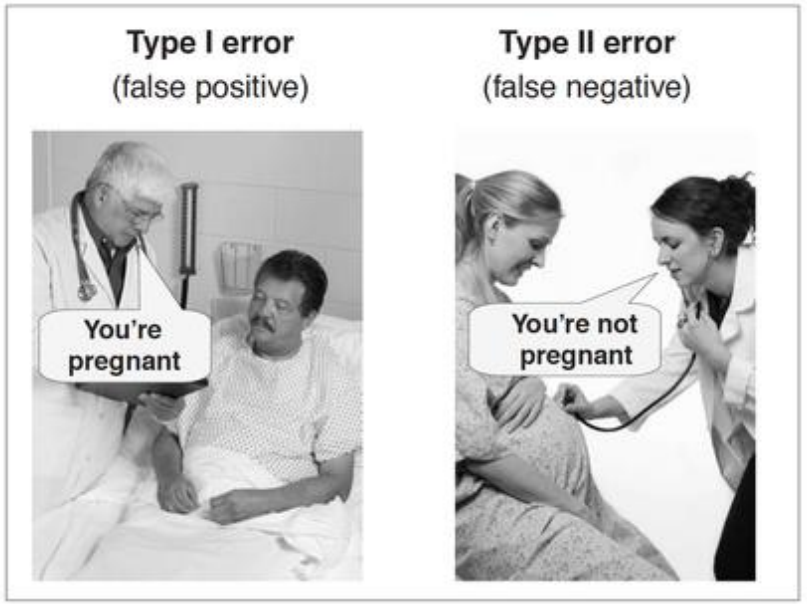
Evaluating a (Binary) Classification Model

- Errors
  - Errors = Misclassification

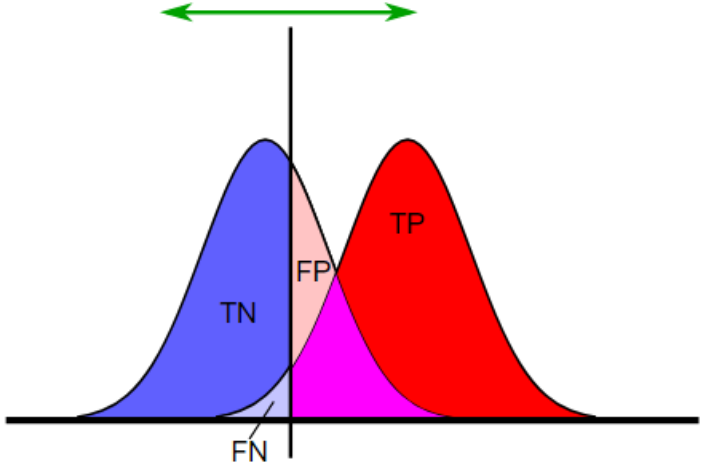
$$\text{Errors} = \sum_{i=1}^{n} \mathrm{I}(y_i \neq \hat{y}_i)$$

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

- Binary: 2 Classes
  - Class-A misclassified as Class-B or
  - Class-B misclassified as Class-A
  - May have different costs
    - Medical Tests, Spam Filtering, Fraud Detection



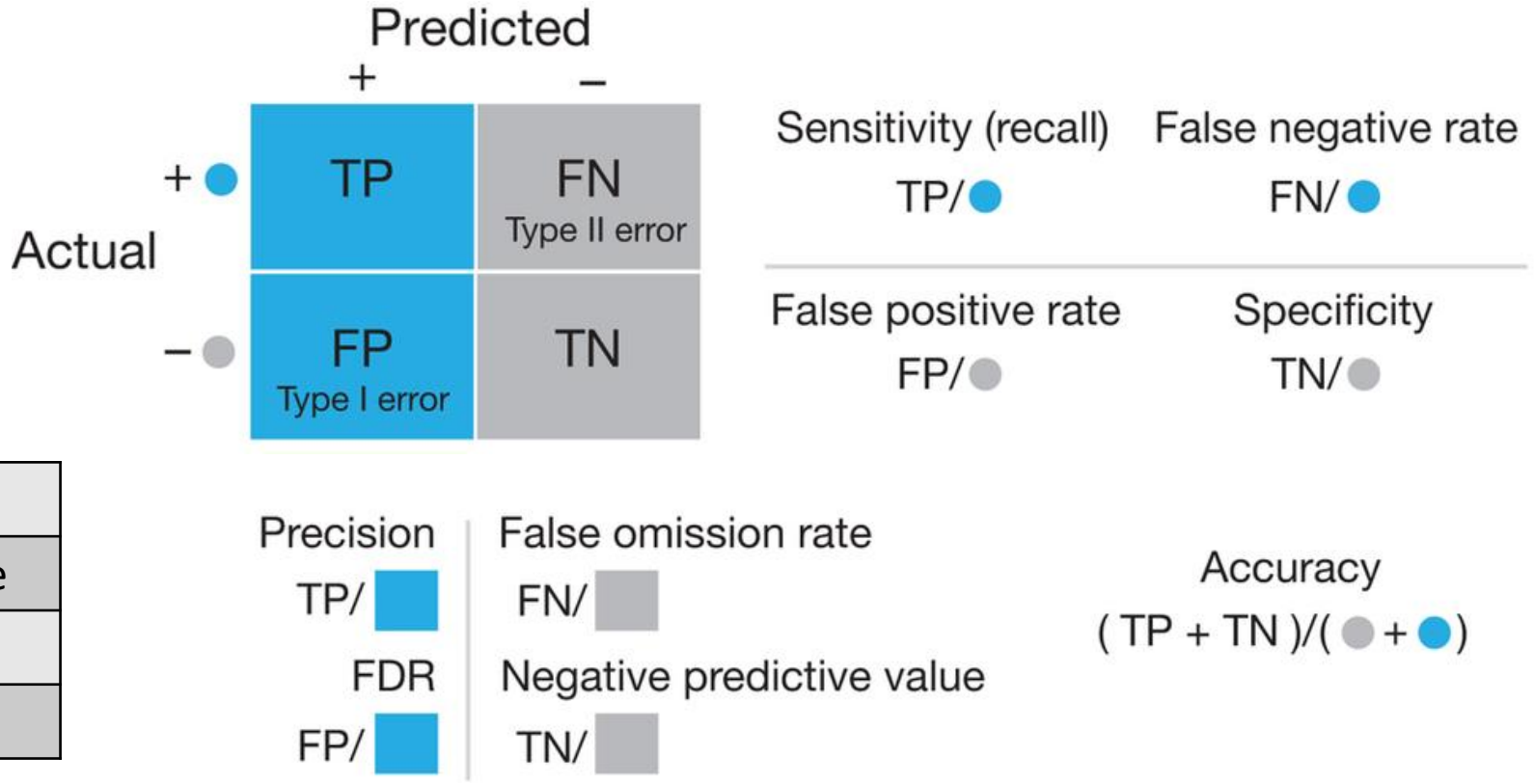
- Recall : Hypothesis Testing
  - Null Hypothesis : Negative Class (Innocent; No Disease)
  - - predicted as + (Type-1 error)
  - + predicted as - (Type-2 error)
  - Tradeoff ←→ Hyperparameter choice



Confusion Matrix : Evaluating a (Binary) Classification Model

- Four Key Measures (Numbers)
  - True Positive
  - True Negative
  - False Positive
  - False Negative
- Confusion Matrix
  - And Derivatives…

10-year CHD risk		Predicted	
		True	False
Actual	True	12	158
	False	7	915



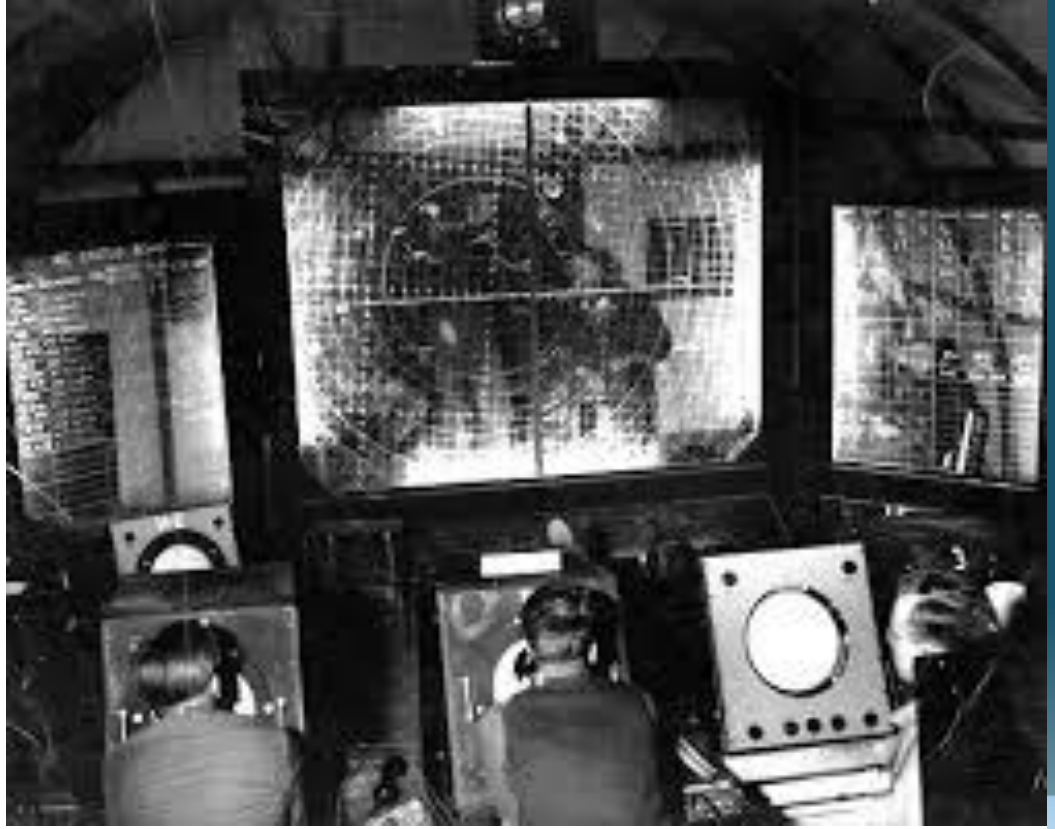
https://www.nature.com/nmeth/journal/v13/n8/fig_tab/nmeth.3945_F1.html

ROC: Receiver Operating Characteristic

- Radar tracking military airplanes

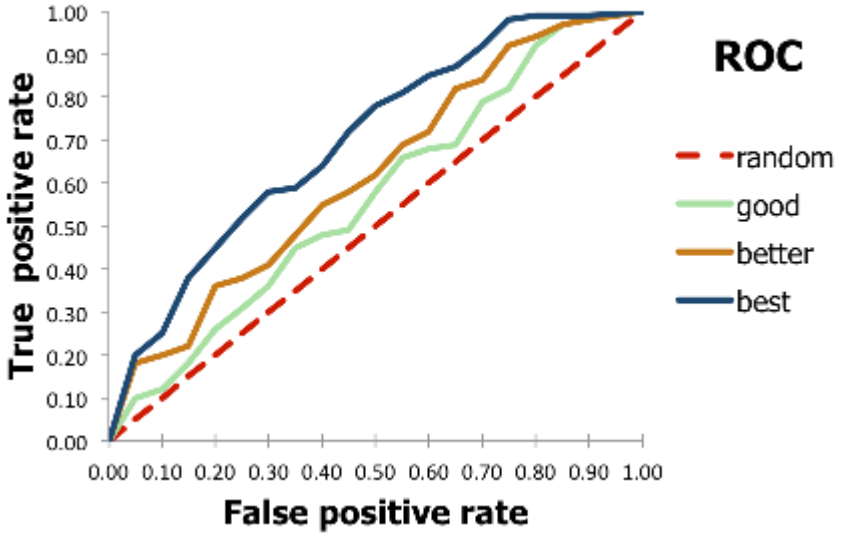
- Detect ➔ Present?
  - True Positive
  - True Negative
  - False Positive
  - False Negative

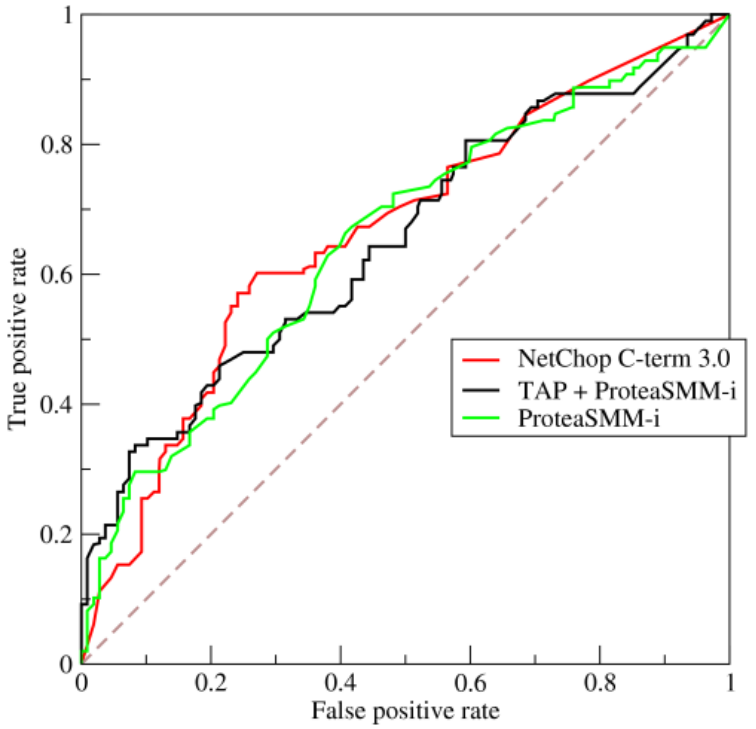
- Tuner (sensitivity)



ROC: Area Under Curve

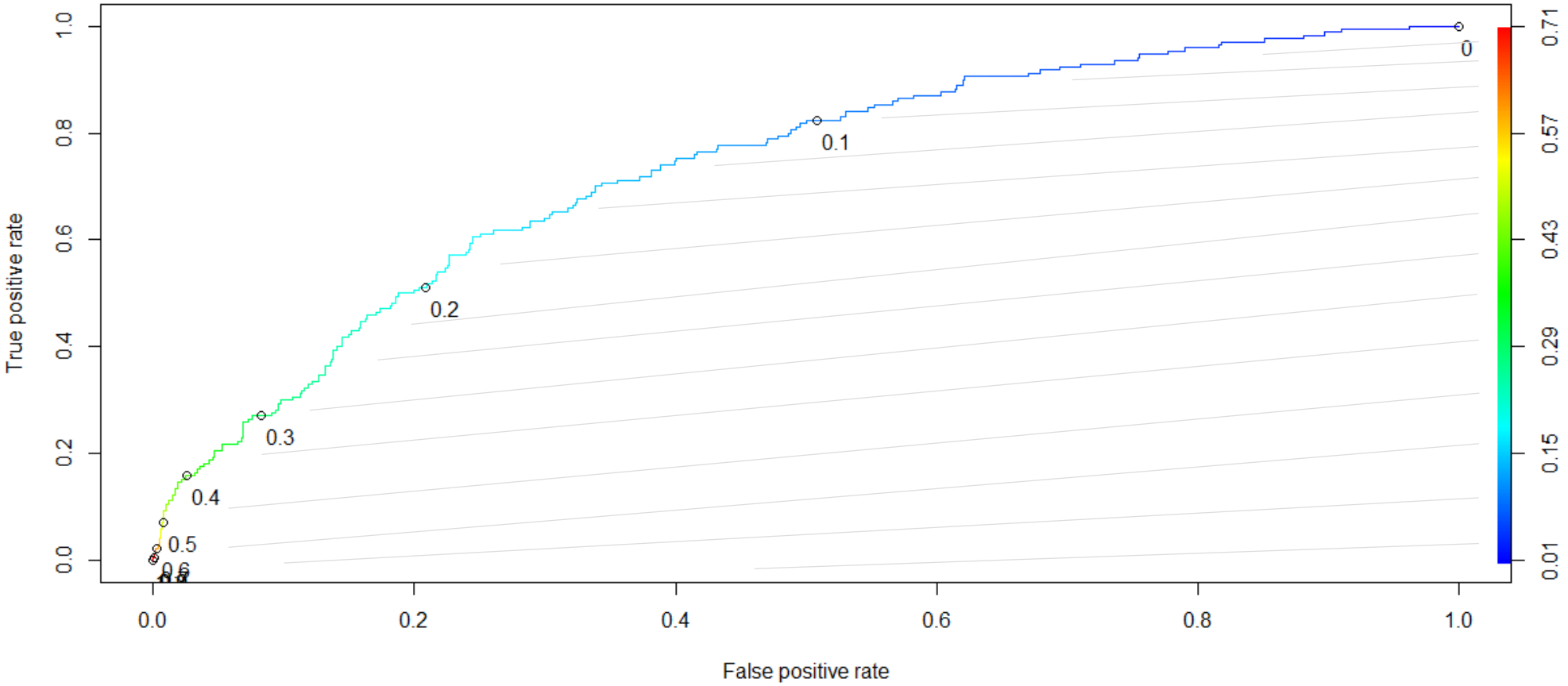
- Measures discrimination
  - Ability to correctly classify
  - Percentage of randomly drawn data for which the classification is done correctly.
- Visual comparison may not suffice
  - Real world classifier performance is “messy”





ROC : Area Under Curve

- AUC = 73.2%



Gain & Lift charts

- Given an observation, a classifier outputs
  - Class = A
  - Pr(Class = A) : Can be used to “rank” observations

- Objective
  - Quantify the business impact of using a classifier (predictive model)
  - Compare business performance with and without (baseline) the predictive model

Example: Gain & Lift charts

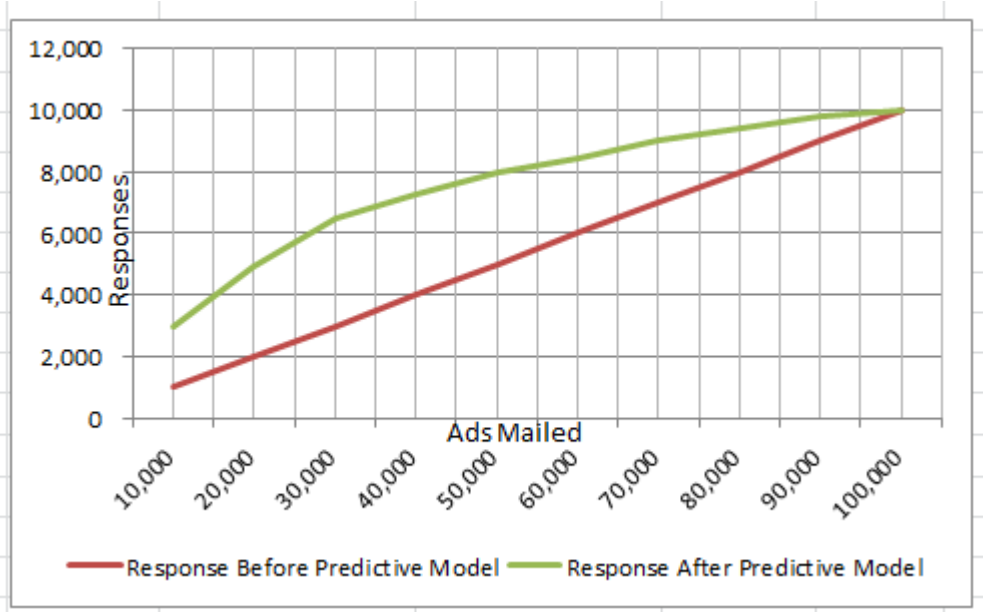
- Marketing Campaign: A company sends mail catalogs to prospective buyers. It costs the company $1 to print and mail one catalog.
  - Baseline: Company randomly contacts the prospects
  - Predictive model: Assign a probability of purchase to each customer
    - Order customers and divided into deciles (or any other quantiles).
    - Contact customers in decreasing order of probability to buy.

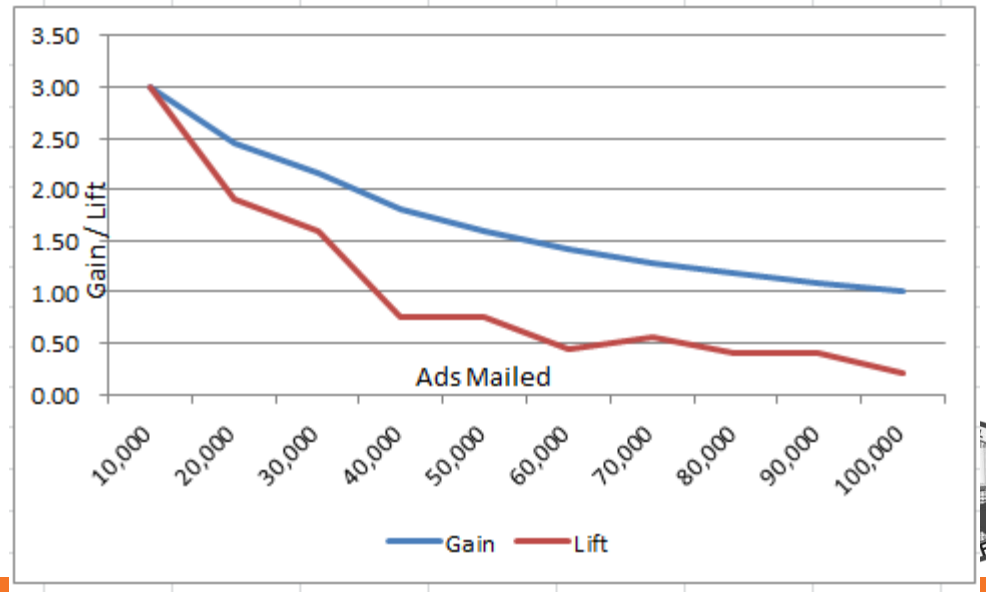
Ads Mailed	Cost @ $1 / Ad Mailed	Response	Return	Profit
10,000	$10,000	1,000	$50,000	$40,000
20,000	$20,000	2,000	$100,000	$80,000
30,000	$30,000	3,000	$150,000	$120,000
40,000	$40,000	4,000	$200,000	$160,000
50,000	$50,000	5,000	$250,000	$200,000
60,000	$60,000	6,000	$300,000	$240,000
70,000	$70,000	7,000	$350,000	$280,000
80,000	$80,000	8,000	$400,000	$320,000
90,000	$90,000	9,000	$450,000	$360,000
100,000	$100,000	10,000	$500,000	$400,000

Ads Mailed	Cost @ $1 / Ad Mailed	Response	Return	Profit
10,000	$10,000	3,000	$150,000	$140,000
20,000	$20,000	4,900	$245,000	$225,000
30,000	$30,000	6,500	$325,000	$295,000
40,000	$40,000	7,250	$362,500	$322,500
50,000	$50,000	8,000	$400,000	$350,000
60,000	$60,000	8,450	$422,500	$362,500
70,000	$70,000	9,000	$450,000	$380,000
80,000	$80,000	9,400	$470,000	$390,000
90,000	$90,000	9,800	$490,000	$400,000
100,000	$100,000	10,000	$500,000	$400,000

Example: Gain & Lift charts (cont'd)

Ads Mailed	Response Before Predictive Model	Response After Predictive Model	Gain	Lift
10,000	1,000	3,000	3.00	3.00
20,000	2,000	4,900	2.45	1.90
30,000	3,000	6,500	2.17	1.60
40,000	4,000	7,250	1.81	0.75
50,000	5,000	8,000	1.60	0.75
60,000	6,000	8,450	1.41	0.45
70,000	7,000	9,000	1.29	0.55
80,000	8,000	9,400	1.18	0.40
90,000	9,000	9,800	1.09	0.40
100,000	10,000	10,000	1.00	0.20





- Gain

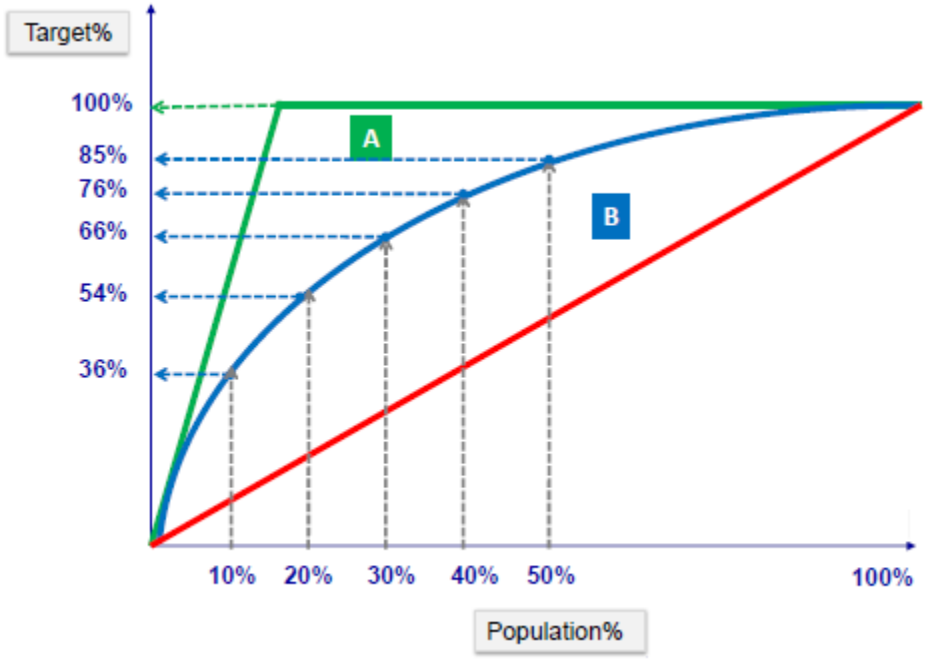
  (Expected Response Using Predictive Model) /
  (Expected Response From Random Mailing)

- Lift

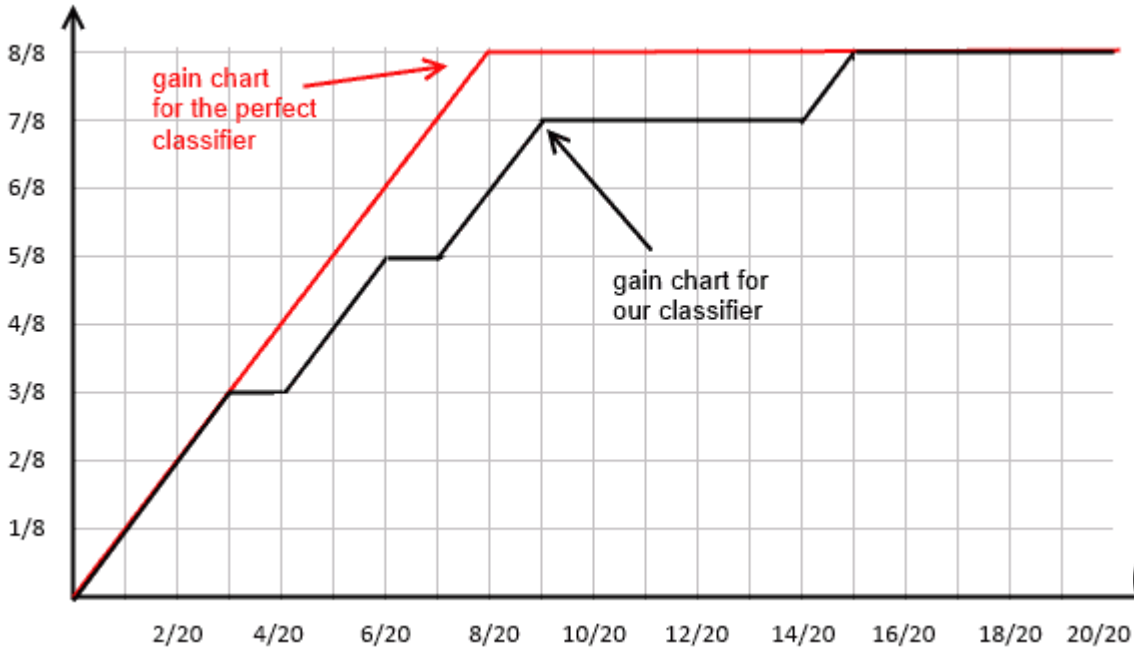
  (Expected Response In A **Specific** Lot Of Prospects Using Predictive Model) /
  (Expected Response In A **Random** Lot Of Prospects Without Using Predictive Model)

Gain & Lift charts

- Objective
  - Quantify the business impact of using a classifier (predictive model)
  - Compare business performance with and without (baseline) the predictive model
- Given an observation, a classifier outputs
  - Class = A
  - Pr(Class = A) : Can be used to "rank" observations e.g. Rank customers in order of Prob. Of purchase
    - Divided into deciles (or any other quantiles).



http://www.saedsayad.com/images/Gain_chart.png

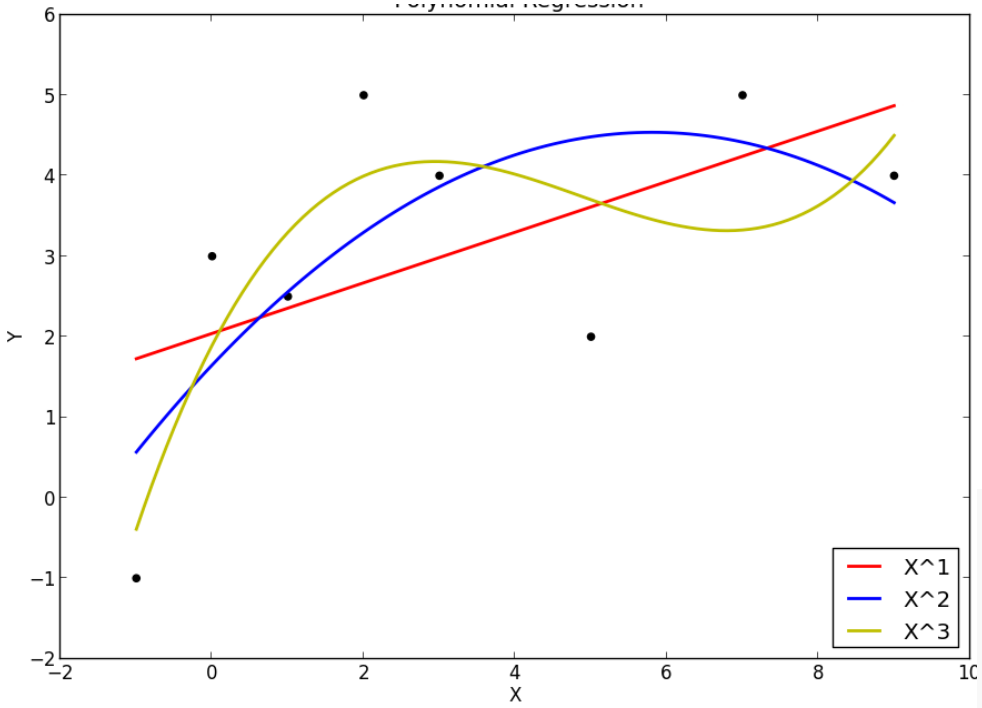


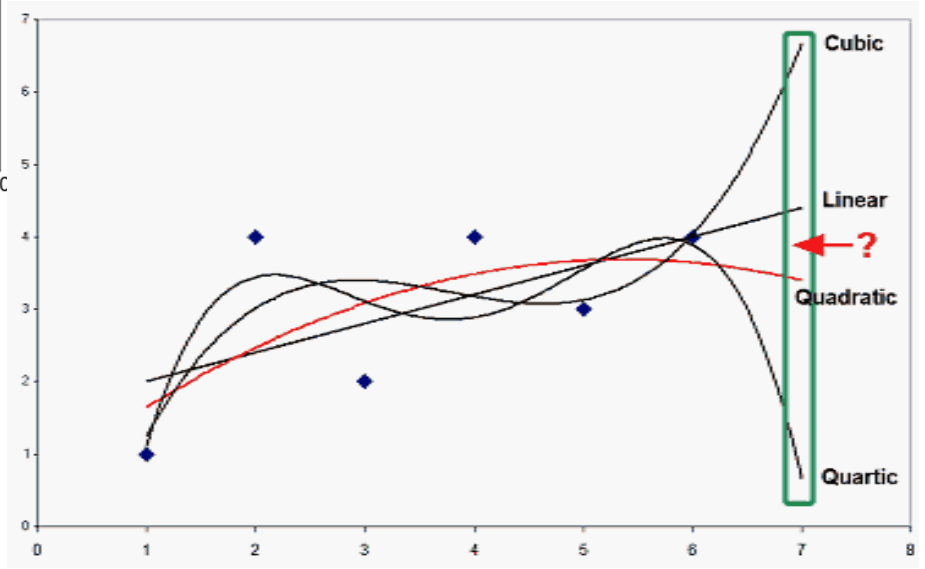
http://mlwiki.org/index.php/Cumulative_Gain_Chart

Comparing Models

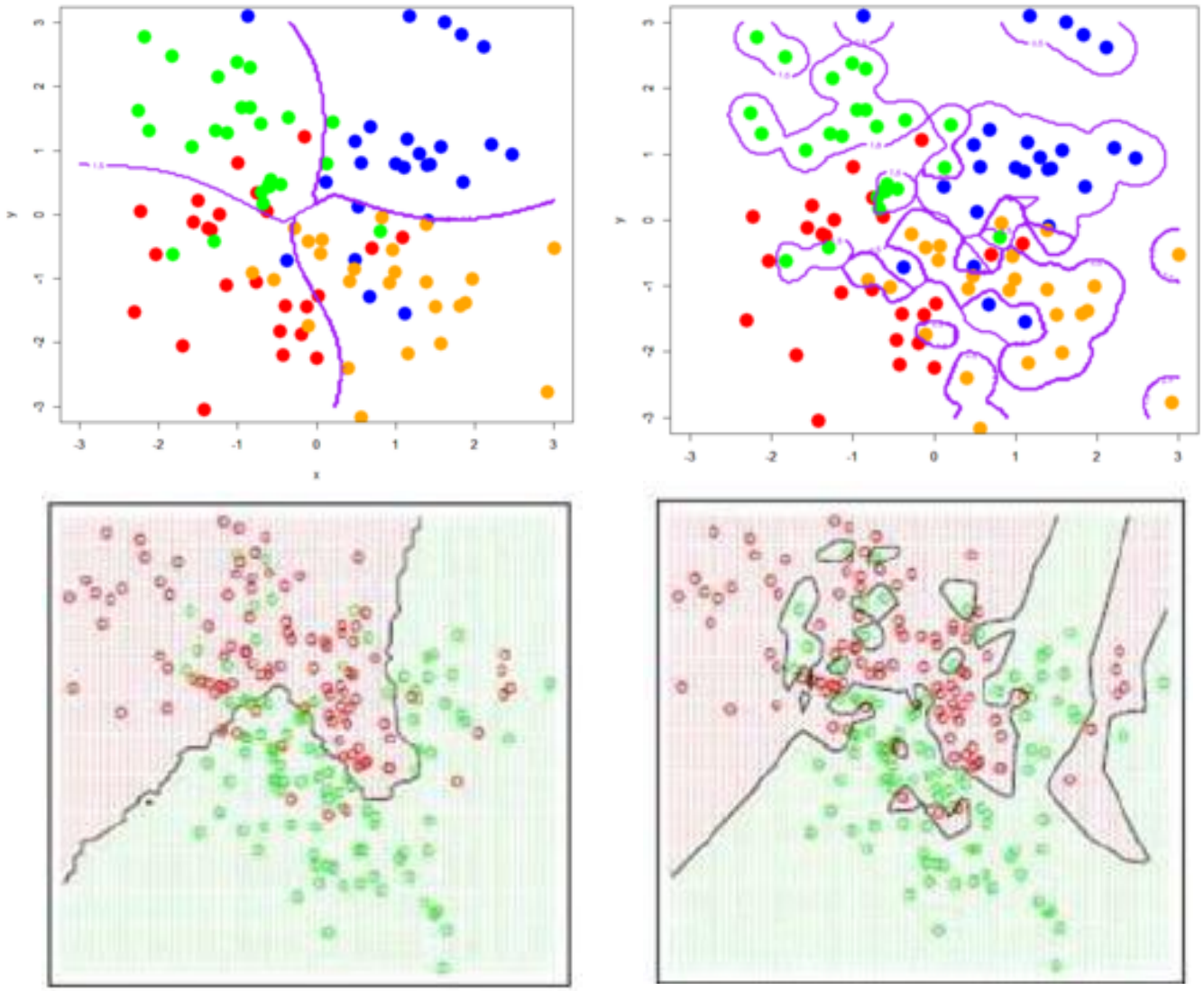
Model Complexity, Bias-Variance, Generalization Error, Overfitting, Hyperparameters vs. Parameters

Reducing error… at what cost? | Regression





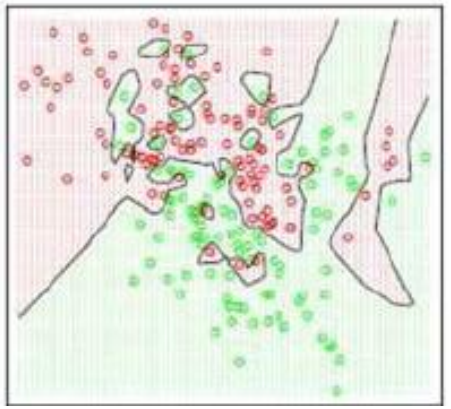
Reducing error… at what cost? | Classification

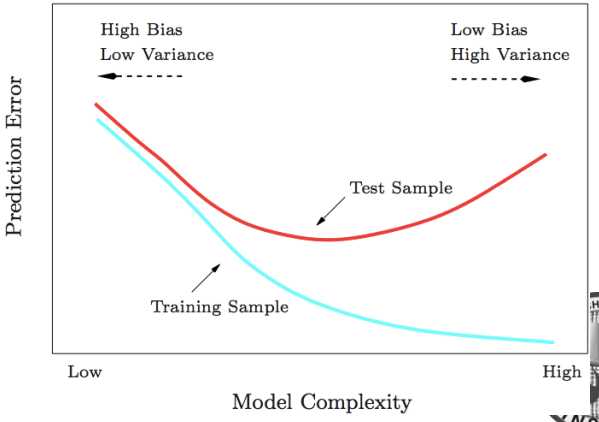


Model Evaluation : Error vs. Complexity

- 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)*
- Approach: 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: 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

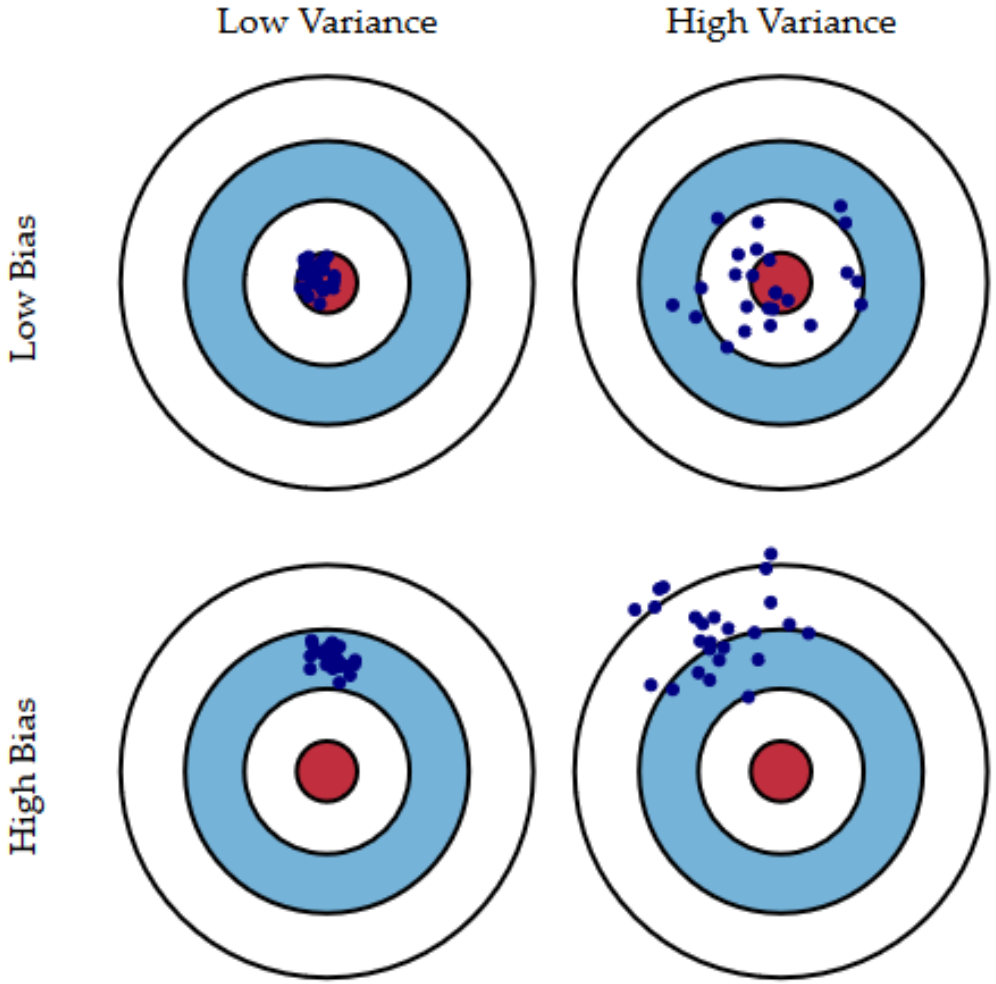




Model Evaluation : Bias vs. Variance tradeoff

- Model Bias
  - Error due to the assumptions (limitations) of the model
  - E.g. linearity, continuous functions.
  - High bias ➔ Look for a different class of functions
    - more “flexible”
    - More complex

- Model Variance
  - How much does the model change with a change in sample?
  - Sensitivity to change in sample (training data)
  - High variance ➔



Bias vs. Variance Tradeoff

- Function Approximation framework
  - Learn a function from the data (which minimizes some error)

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

- Error
  - Depends on the sample
  - Depends on the choice of the model family

$$\mathrm{E}\left[\left(y - \hat{f}(x)\right)^2\right]$$

$$\mathrm{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathrm{Bias}\left[\hat{f}(x)\right]^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

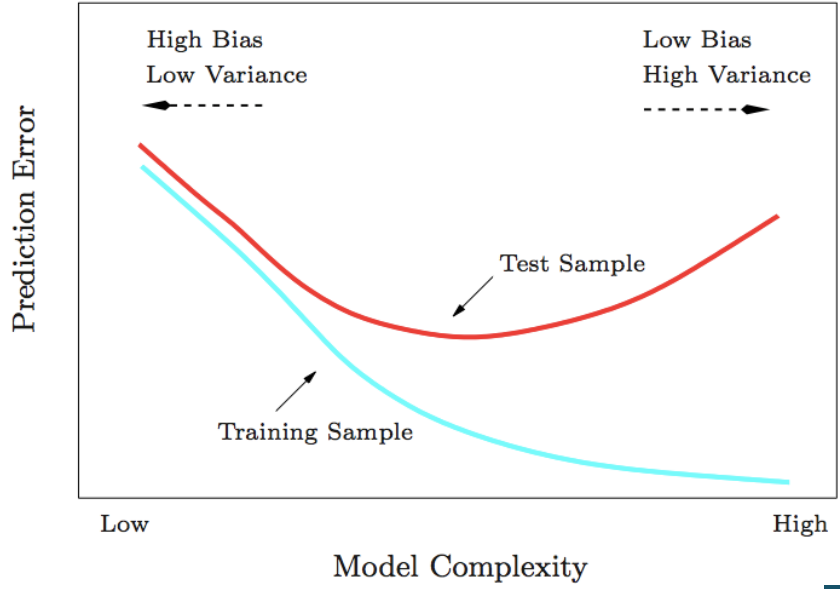
- Bias-vs-Variance Tradeoff
  - Increase complexity to reduce bias
  - ➔ Make it more sensitive to the data
  - ➔ Make it more sensitive to the training data (sample)
  - ➔ Increase Variance

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathrm{E}\left[\hat{f}(x) - f(x)\right]$$

$$\mathrm{Var}\left[\hat{f}(x)\right] = \mathrm{E}[\hat{f}(x)^2] - \mathrm{E}[\hat{f}(x)]^2$$

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)

Complexity-aware Model Evaluation

Validation Set

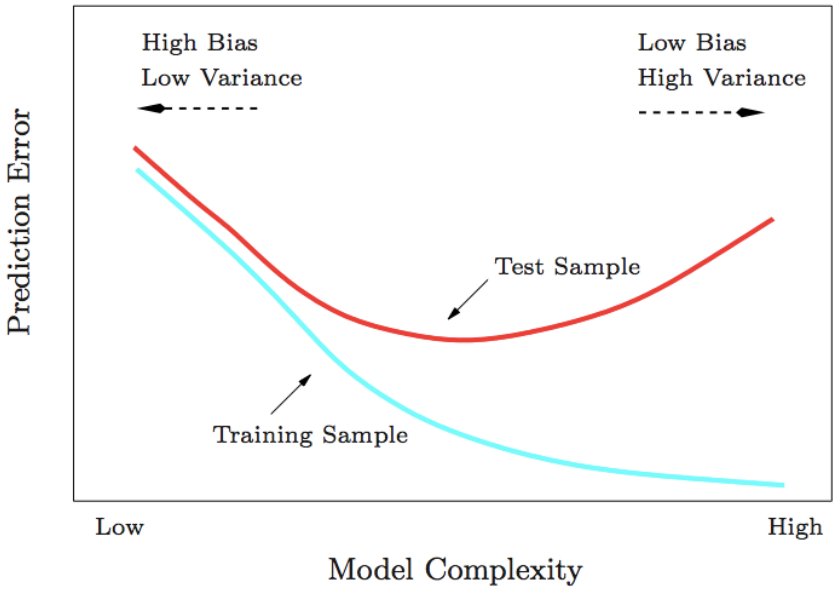
- Key Idea : Assume you have less data available than you actually have
- Split your data into training & test (validation)
- Learn the model on training set. Evaluate (Test) it on validation

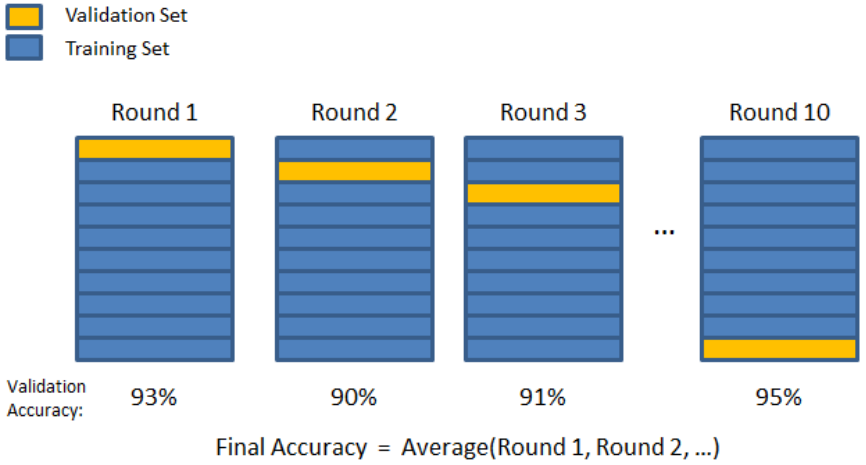
LOOCV

- 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





Q?

Praphul Chandra

Statistical Decision Theory

Praphul Chandra

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

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

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

- 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**

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

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

The best prediction of Y at an point X=x is….

$$\text{Loss Function} : \sum_{i=1}^{n} L(y_i, c) = \sum_{i=1}^{n} (y_i - c)^2$$

$$\text{Minimize Loss} : \frac{d}{dc} \sum_{i=1}^{n} (y_i - c)^2 = 0$$

$$-1 \times 2 \times \sum_{i=1}^{n} (y_i - c) = 0 \Rightarrow \sum_{i=1}^{n} (y_i - c) = 0$$

$$\sum_{i=1}^{n} y_i = nc \Rightarrow c = \frac{1}{n} \sum_{i=1}^{n} y_i$$

c is the mean of y_i

$$\text{Loss Function} : \sum_{i=1}^{n} L(y_i, c) = \sum_{i=1}^{n} |y_i - c|$$

$$\text{Minimize Loss} : \frac{d}{dc} \sum_{i=1}^{n} |y_i - c| = 0$$

$$-sign \sum_{i=1}^{n} |y_i - c| = 0$$

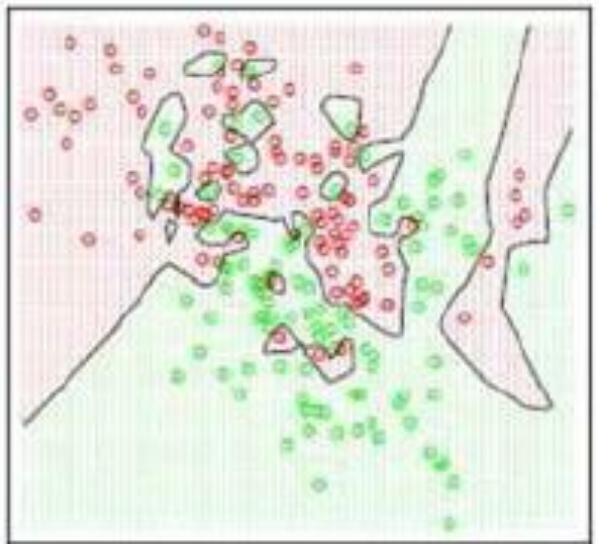
Derivate vanishes when there is the same number of positive and negative terms among the y_i - c which (roughly speaking) arises when cis the median of the y_i.

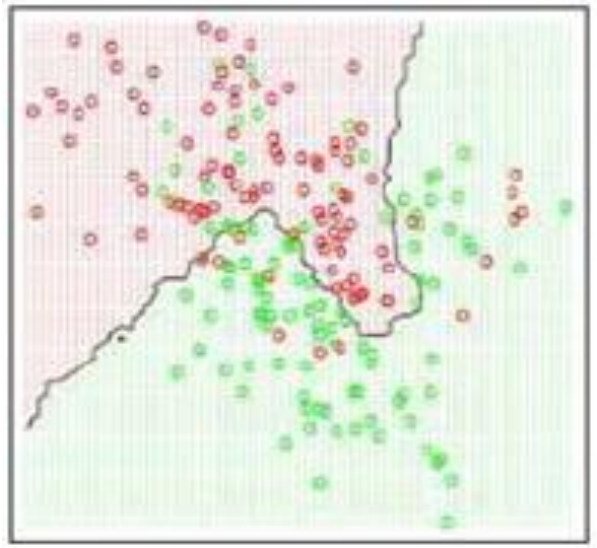
K-Nearest Neighbor

- 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
- 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
  - Model Free *(No assumption on form of f)*
  - Computational Complexity *(Time, Space)*
  - Locally constant
- Behavior
  - Large k : Smoother boundaries
  - Large N : Large storage req. (space complexity)
  - Large p : lower accuracy (curse of dimensionality)

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

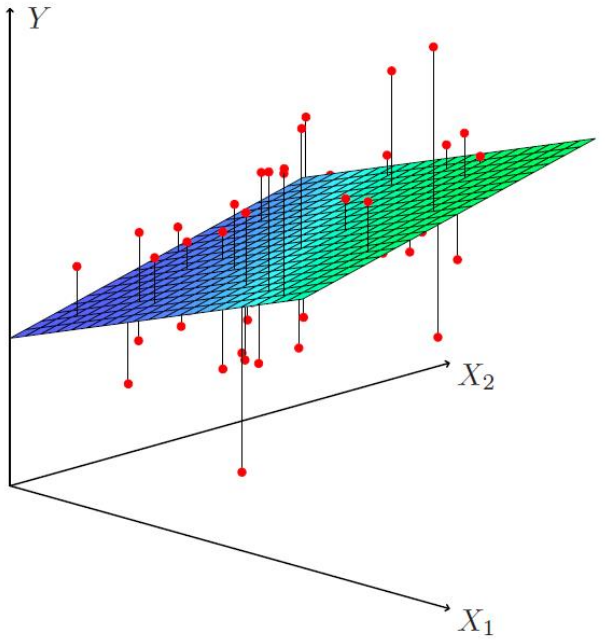
$$\hat{f}(x) = Ave(y_i|x_i \in N_k(x))$$





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)*

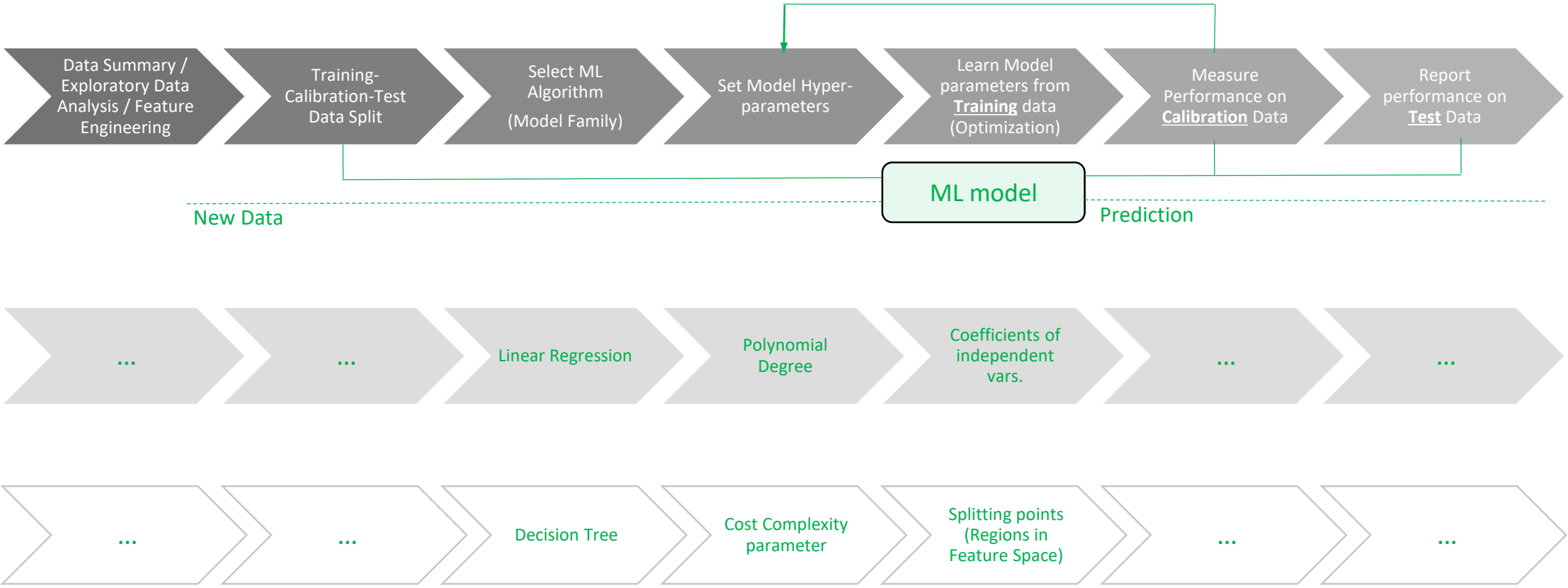


Statistical Decision Theory: Summary $Y = f(X)$

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

Machine Learning Framework



Improving (Speeding up) knn

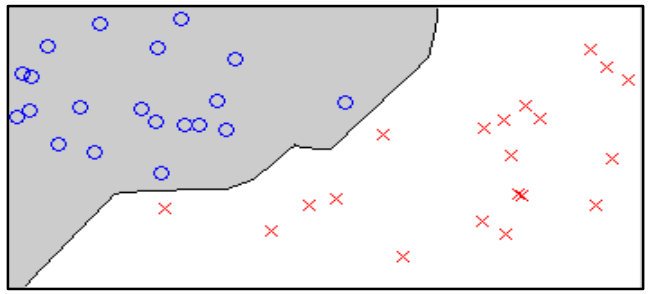
- Clustering as a pre-processing step
  - Eliminate most points (keep only cluster centroids)
  - Apply knn

- Condensed nn
  - Retain samples closest to “decision boundaries”
  - Decision Boundary Consistent – a subset whose nearest neighbour decision boundary is identical to the boundary of the entire training set
  - Minimum Consistent Set – the smallest subset of the training data that correctly classifies all of the original training data

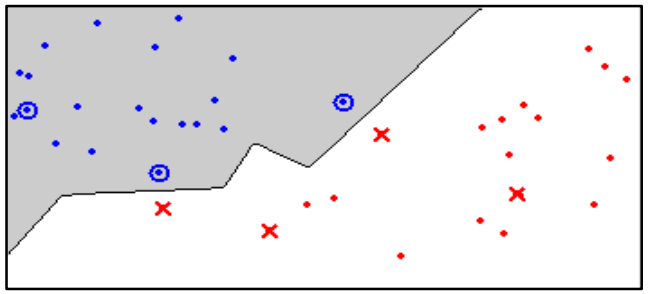
- Reduced nn
  - Remove a sample if doing so does not cause any incorrect classifications

1. Initialize subset with a single training example
2. Classify all remaining samples using the subset, and transfer any incorrectly classified samples to the subset
3. Return to 2 until no transfers occurred or the subset is full

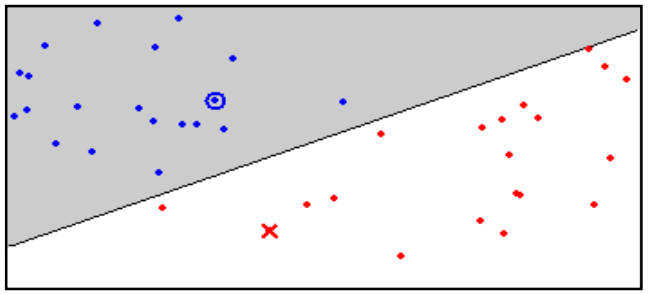
Cran library: Class



Original data



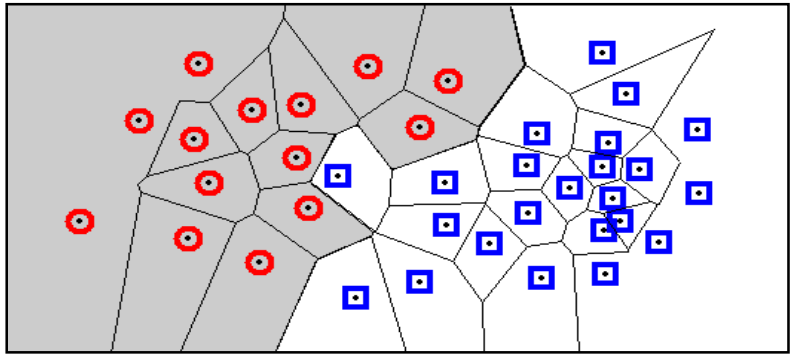
Condensed data



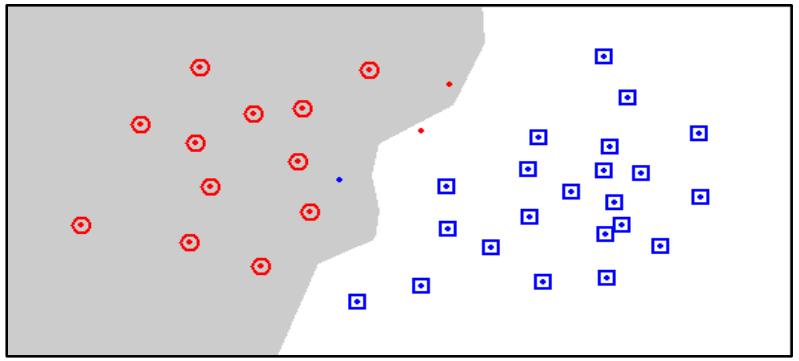
Minimum Consistent Set

Improving (Smoothing) knn | Avoid overfitting

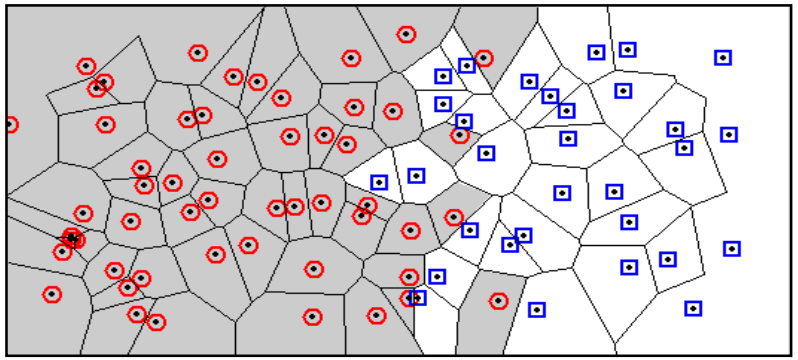
- Wilson editing : Remove points that do not agree with the majority of their k nearest neighbours
- Edited NN



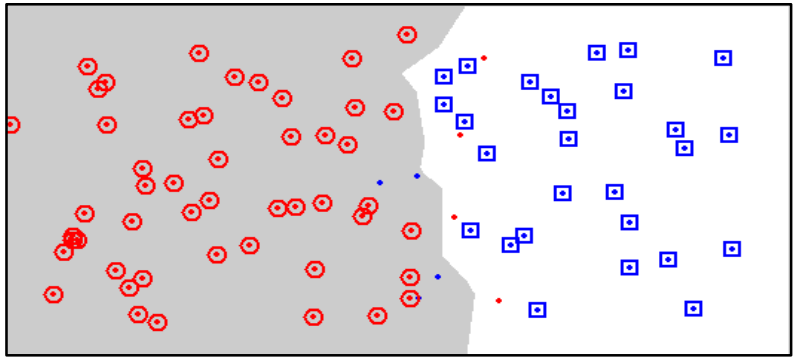
Original data



Wilson editing with k=7



Original data



Wilson editing with k=7

knn: Summary

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

- Lazy | Model Free *(No assumption on form of f)*
- Computational Complexity *(Time, Space)*

- Distance based algorithm
  - Scaling attributes is important
  - Attributes with larger range can dominate e.g., Age versus Salary
  - May not be suitable for high dimensional data
- Categorical variables and Ordinal variables need to be appropriately measured in distance
  - Think distance w.r.t the target