CS 405/605 Data Science

Dr. Qianqian Tong

Course Schedule

Week 13: April 3 topic: Random Forest
April 5 topic: Validation
Project stage IV and V released, and the ddl will be April 28.

Week 14: April 10 topic: PCA
April 12 topic: Clustering-Kmeans

Week 15: April 17 topic: Visualization
April 19 topic: Visualization
HW3 will be released, and the ddl will be April 28.

Week 16: April 24 Project Presentation (4 groups, each will have 15-20 min)
April 26 Project Presentation (4 groups, each will have 15-20 min)
All reports and homework must be submitted by April 28, and graded by the final week.

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Metrics for Performance Evaluation

- Regression
  - Sum of squares

$$\frac{1}{N}\sum_{i=N}^{N}(y_i - f(\mathbf{x}_i))^2$$

  - Sum of deviation

$$\frac{1}{N}\sum_{i=N}^{N}|y_i - f(\mathbf{x}_i)|$$

  - Coefficient of determination R^2

$$1 - \frac{\sum_i (y_i - f(\mathbf{x}_i))^2}{\sum_i (y_i - \bar{y})^2}$$

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Metrics for Performance Evaluation

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)

Metrics for Performance Evaluation

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	a (TP)	b (FN)
ACTUAL CLASS Class=No	c (FP)	d (TN)

- F-measure is biased towards all except C(No|No)

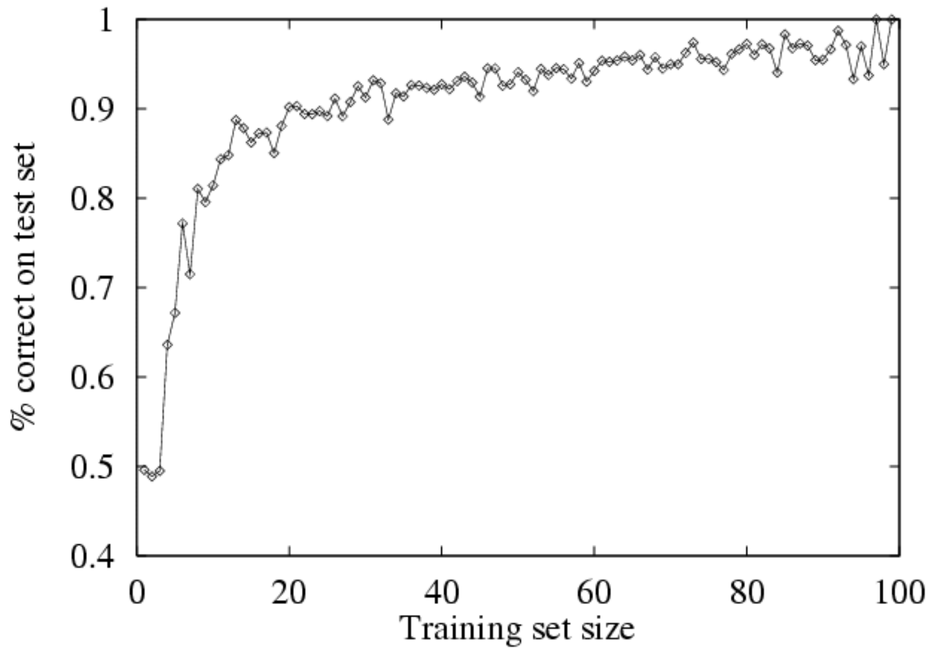
$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Validation

Assessing Performance

problem

- simulate 100 data sets of different sizes
- train on this data, and assess performance on an independent test set
- learning curve = plotting accuracy as a function of training set size
- typical "diminishing returns" effect (some nice theory to explain this)

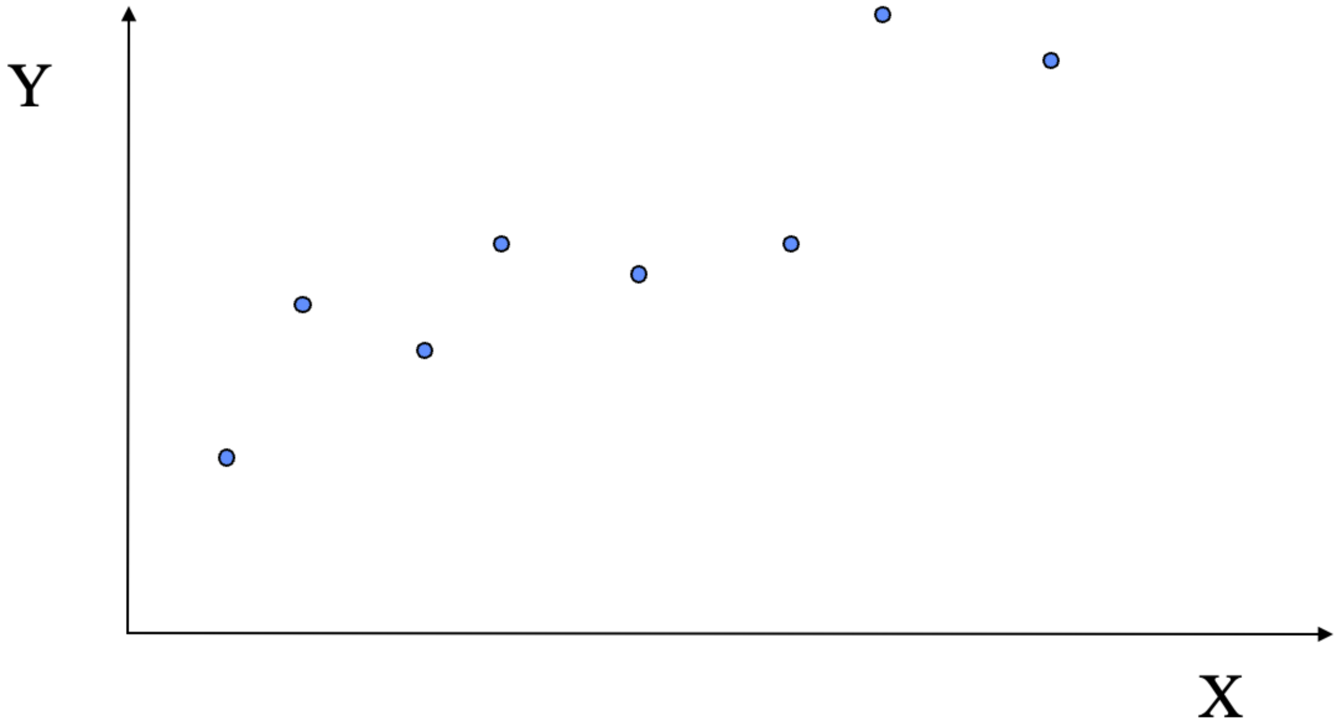


UNC GREENSBORO

Validation

Assessing Performance

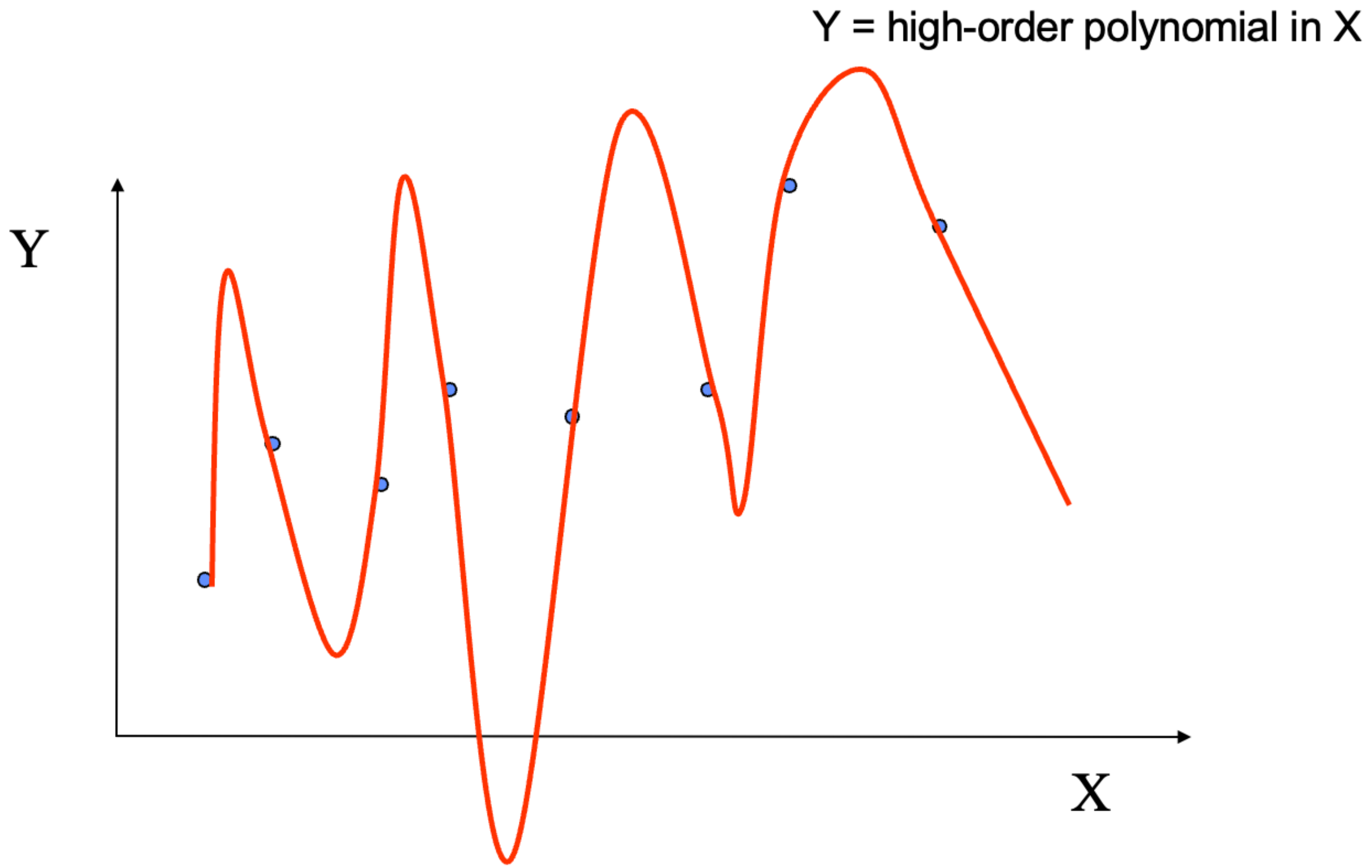
Overfitting and Underfitting



Assessing Performance

Overfitting and Underfitting

A Complex Model

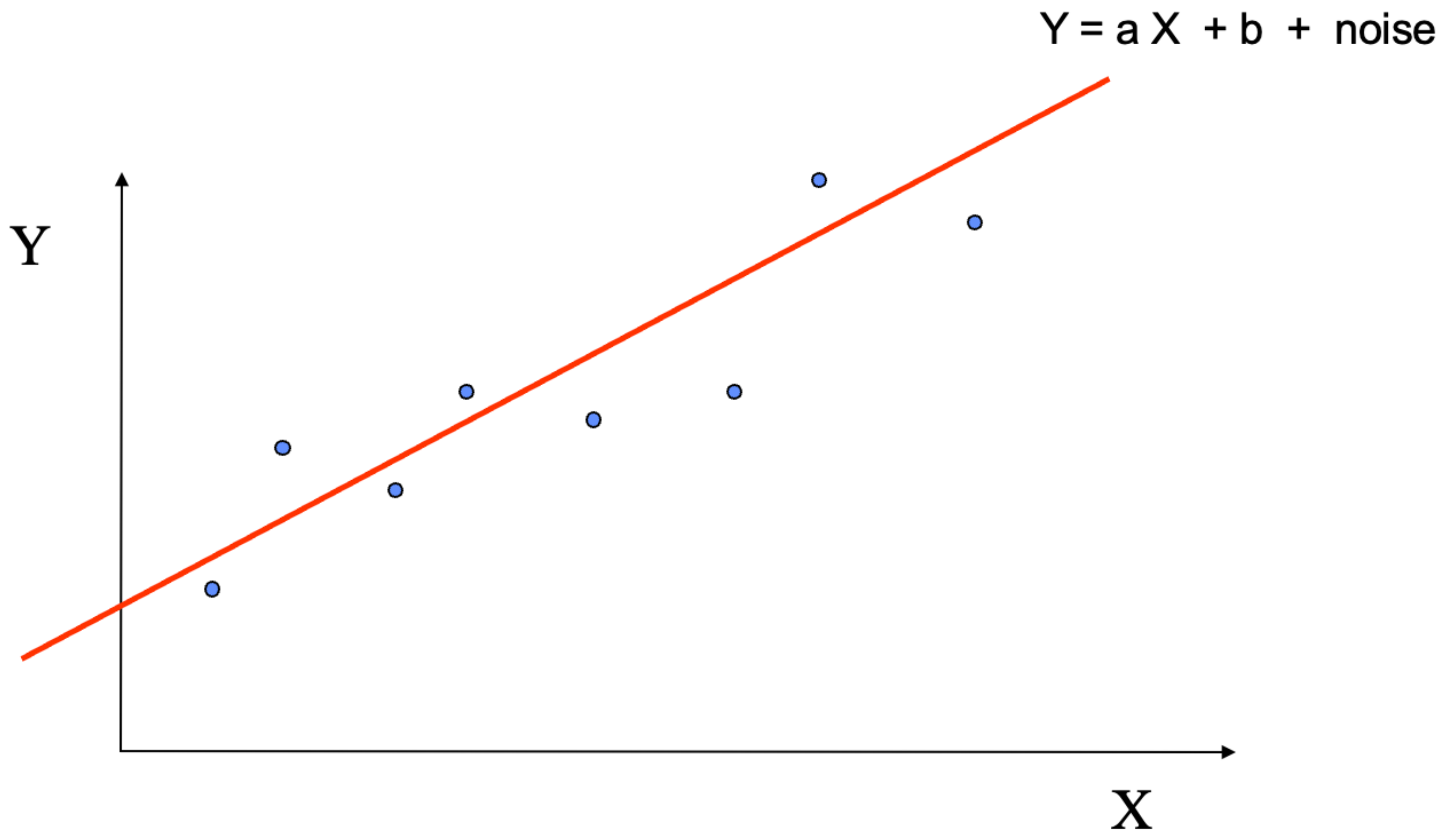


Validation

Assessing Performance

Overfitting and Underfitting

A Simple Model

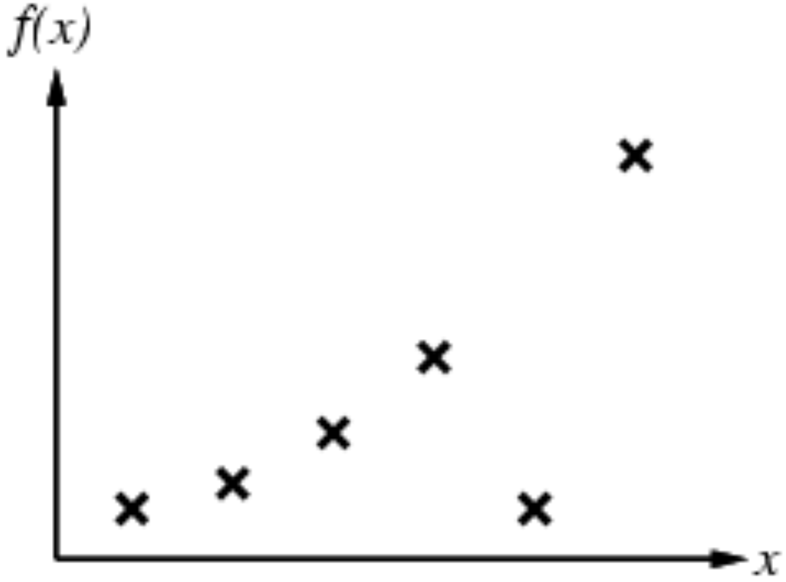


UNC GREENSBORO

Assessing Performance

Overfitting and Underfitting

Another example

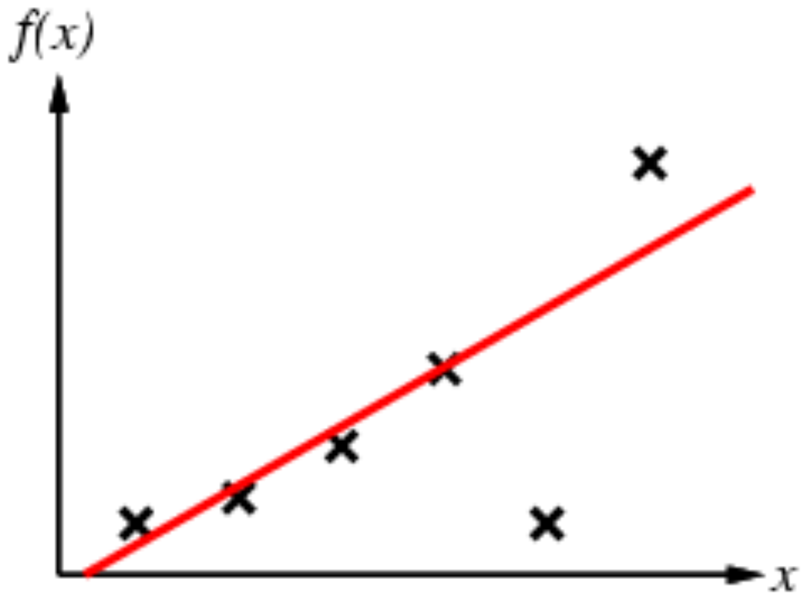


Validation

Assessing Performance

Overfitting and Underfitting

Simple linear model

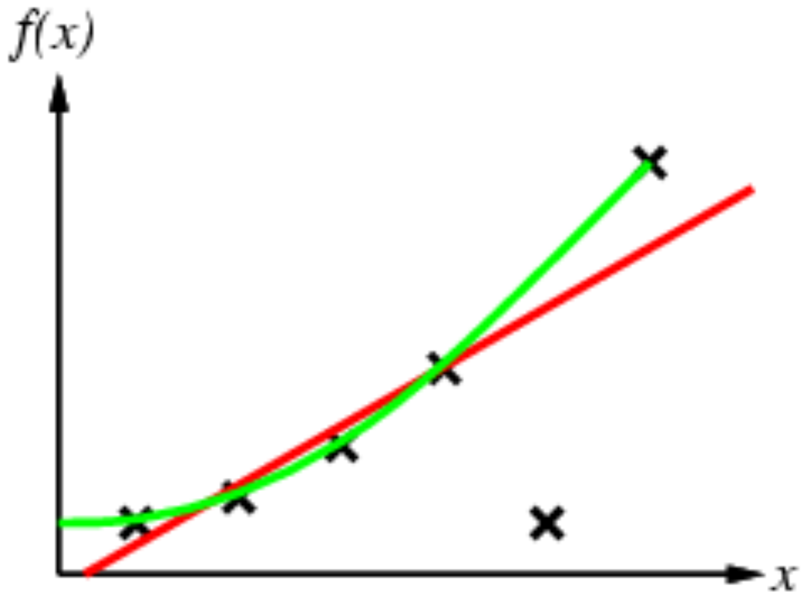


Validation

Assessing Performance

Overfitting and Underfitting

Simple linear model
VS
High order model

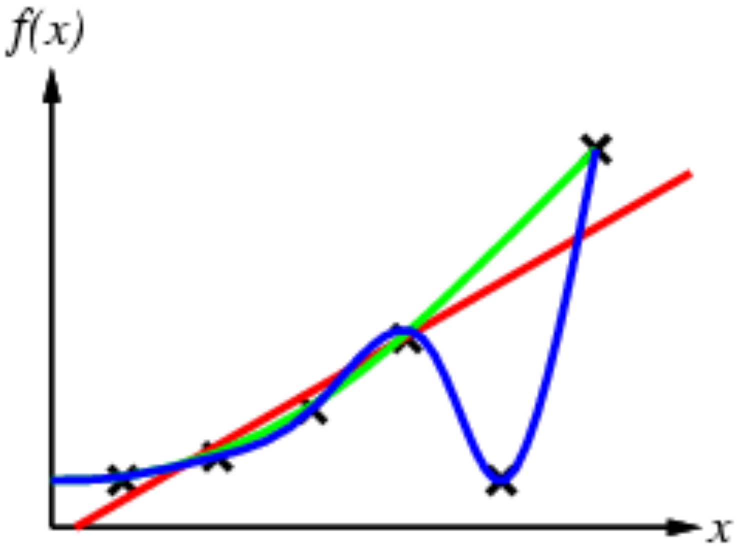


Validation

Assessing Performance

Overfitting and Underfitting

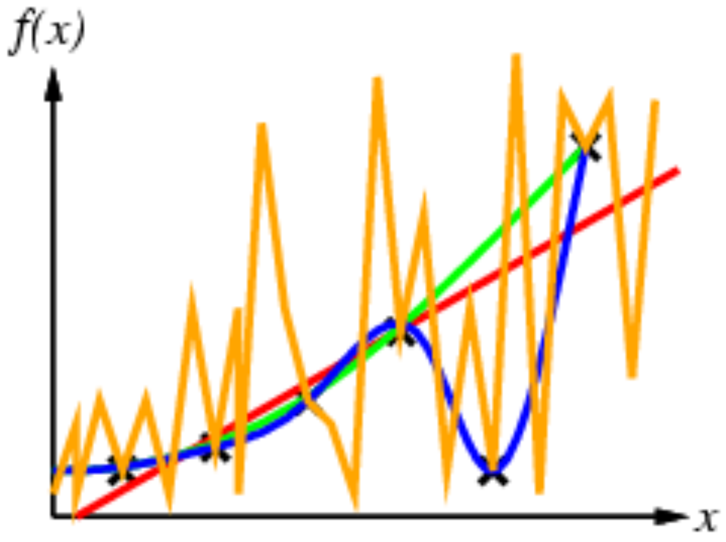
Simple linear model
VS
High order model



Assessing Performance

Overfitting and Underfitting

Simple linear model
VS
High order model

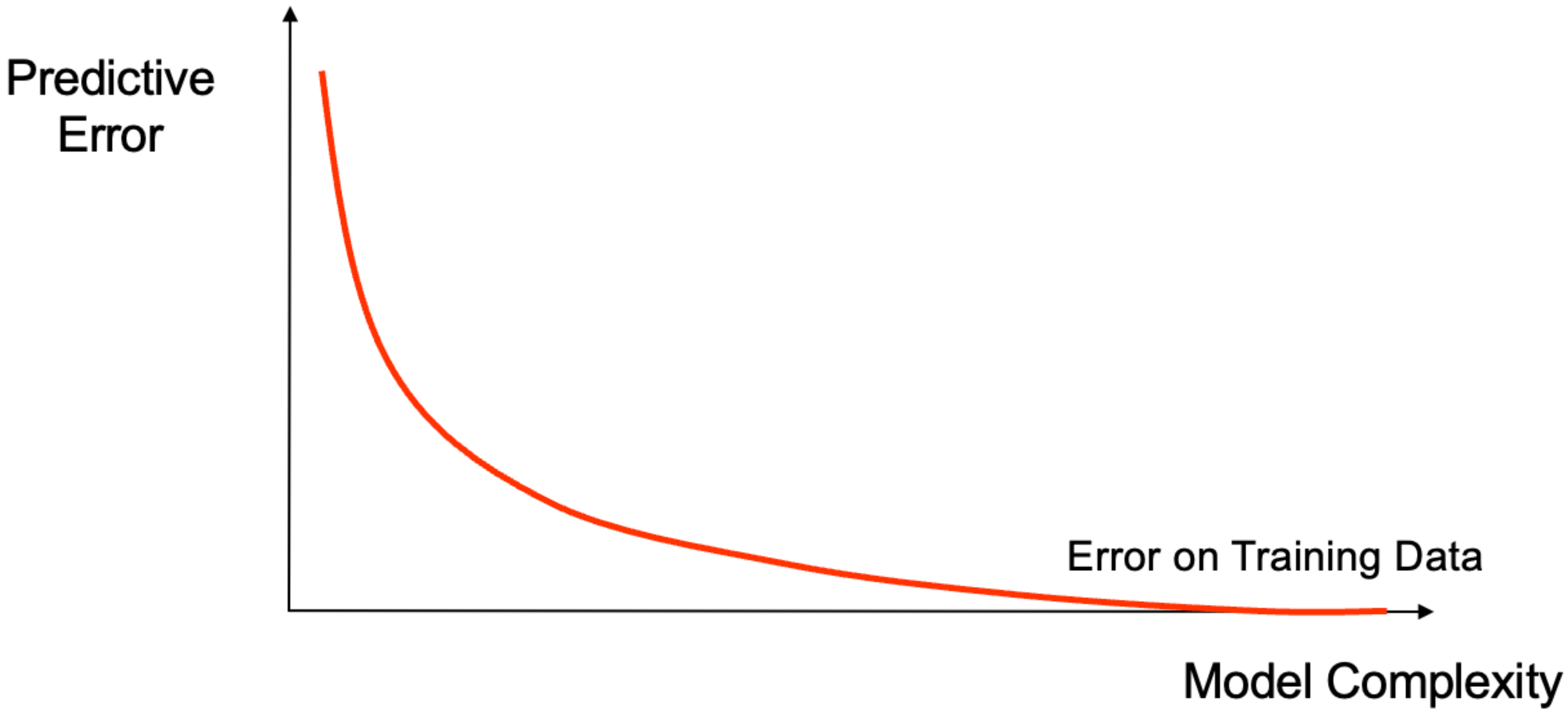


Validation

Assessing Performance

Overfitting and Underfitting

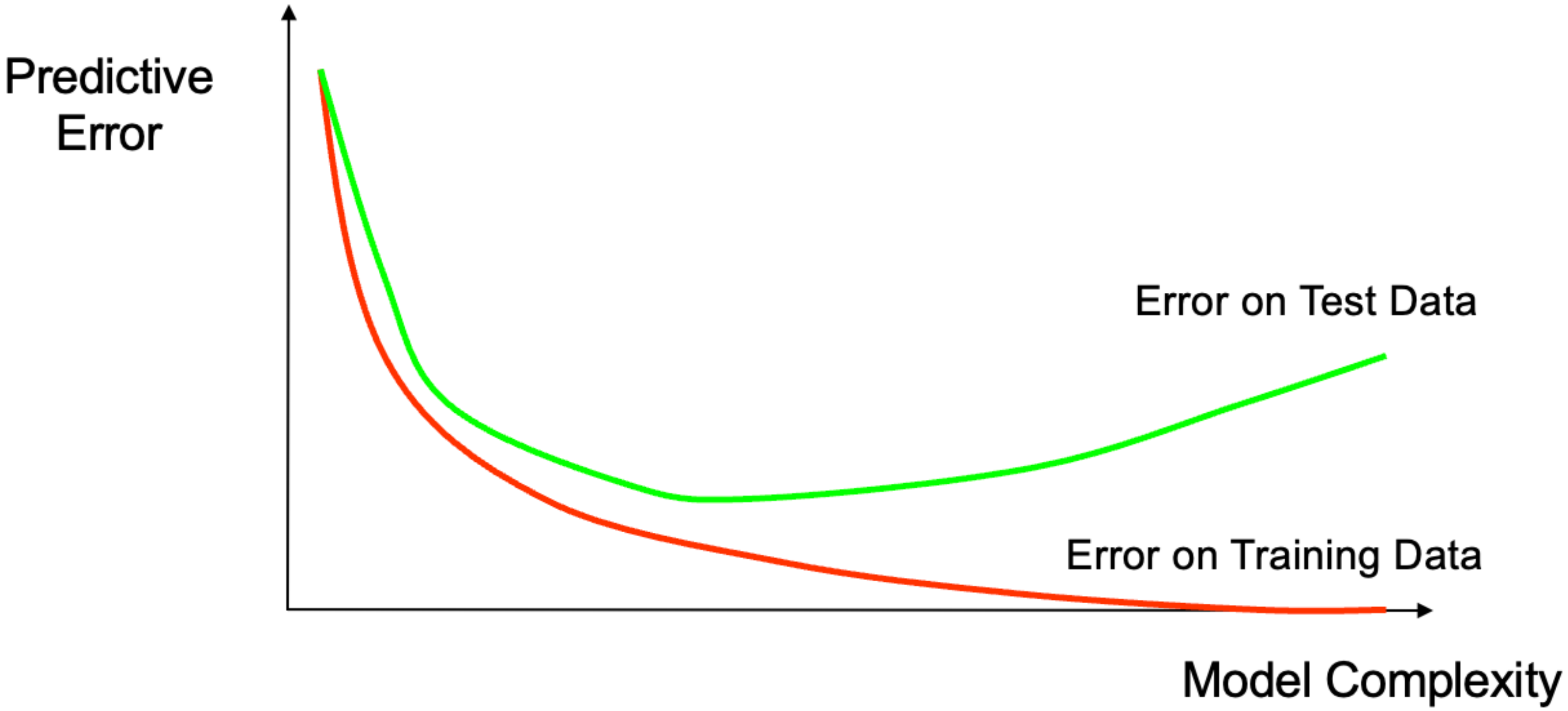
How Overfitting affects Prediction



Assessing Performance

Overfitting and Underfitting

How Overfitting affects Prediction

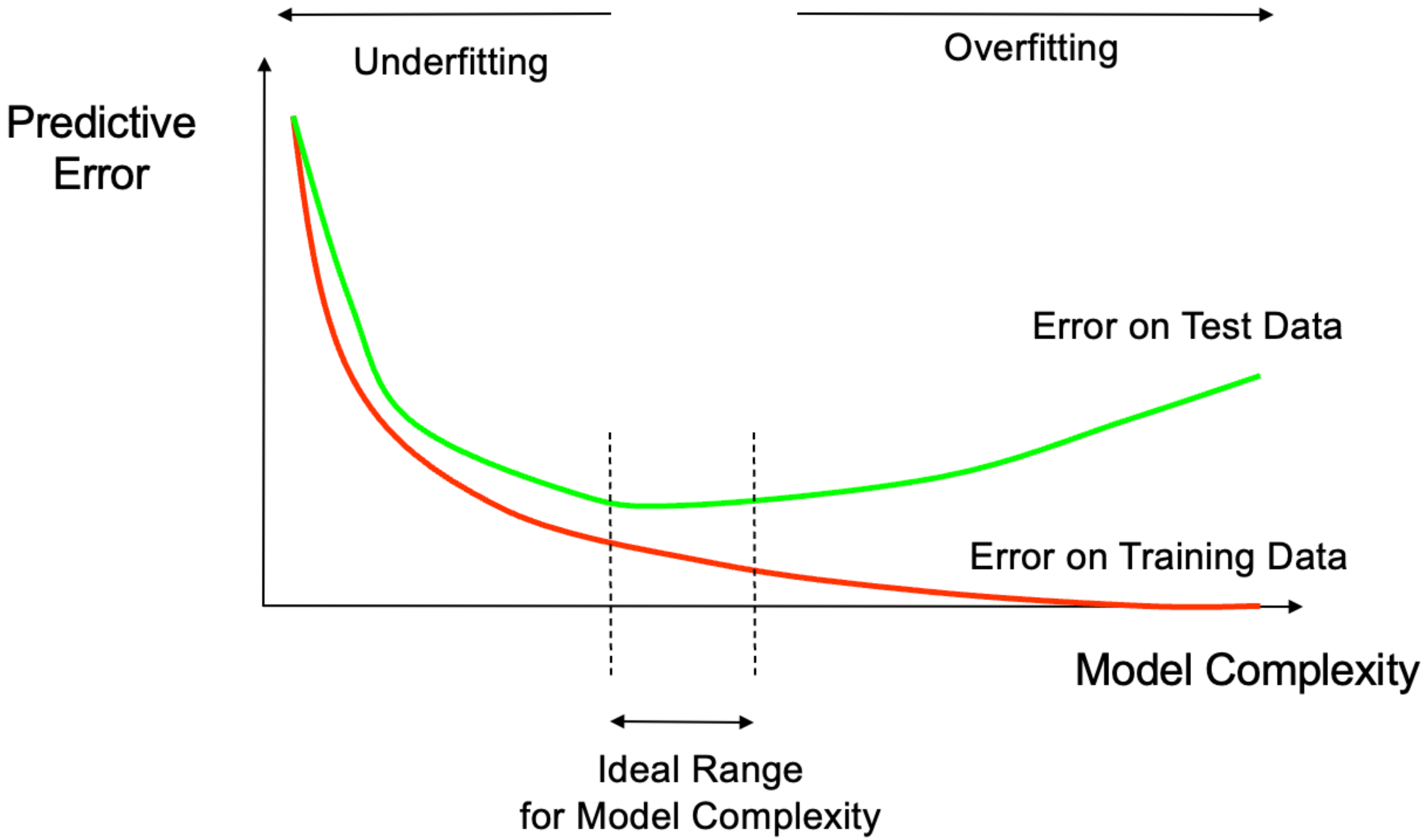


Validation

Assessing Performance

Overfitting and Underfitting

How Overfitting affects Prediction



Model Evaluation

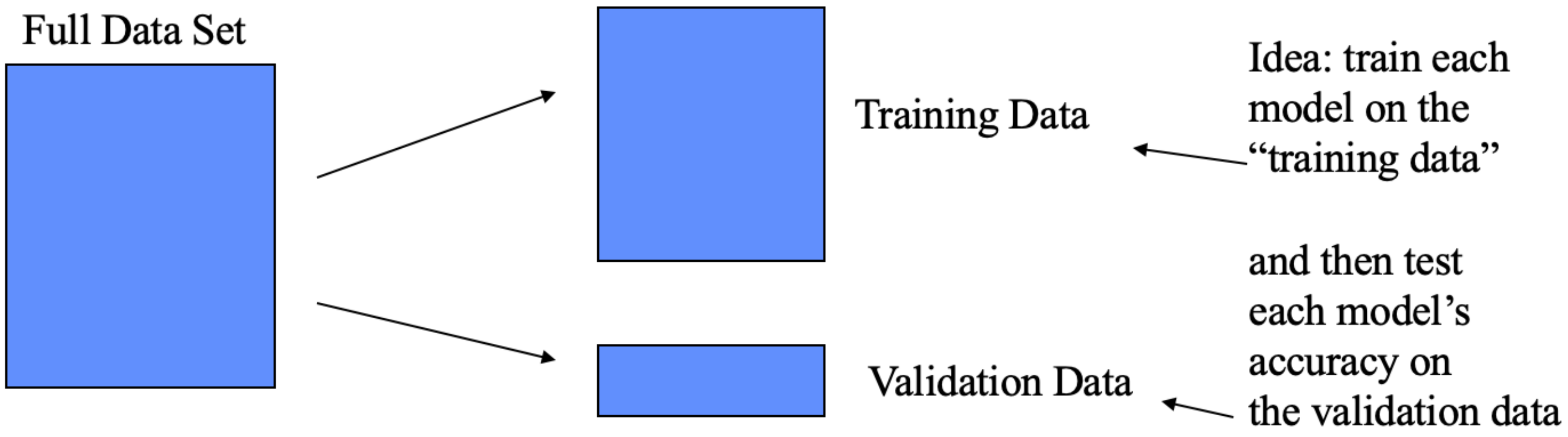
- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Methods for Performance Evaluation

- Holdout
  - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
  - Repeated holdout
- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n
- Stratified sampling
  - oversampling vs undersampling
- Bootstrap
  - Sampling with replacement

Methods for Performance Evaluation

- Holdout method
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation



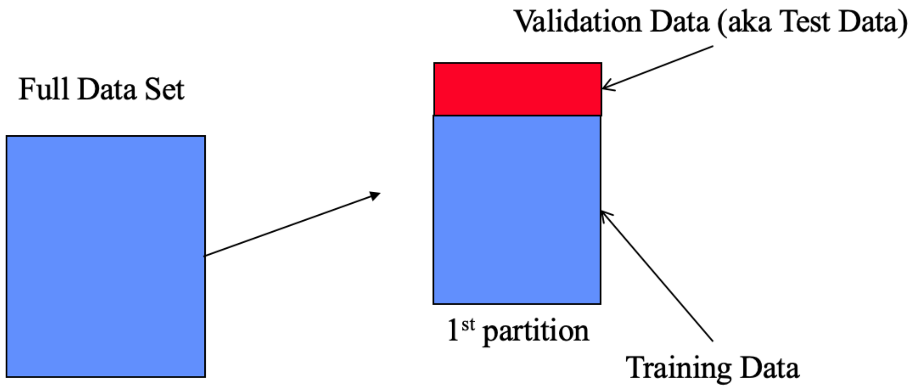
Methods for Performance Evaluation

- Cross-validation (*k*-fold, where k = 10 is most popular)
  - randomly partition our full data set into k disjoint subsets (each roughly of size n/v, n = total number of training data points)
    - for i = 1:10 (here k = 10)
      - train on 90% of data,
      - Acc(i) = accuracy on other 10%
    - end
    - Cross-Validation-Accuracy = 1/k $\sum_i$ Acc(i) choose the method with the highest cross-validation accuracy
  - common values for k are 5 and 10
  - Can also do “leave-one-out” where k = n

Methods for Performance Evaluation

- Cross-validation (*k*-fold)

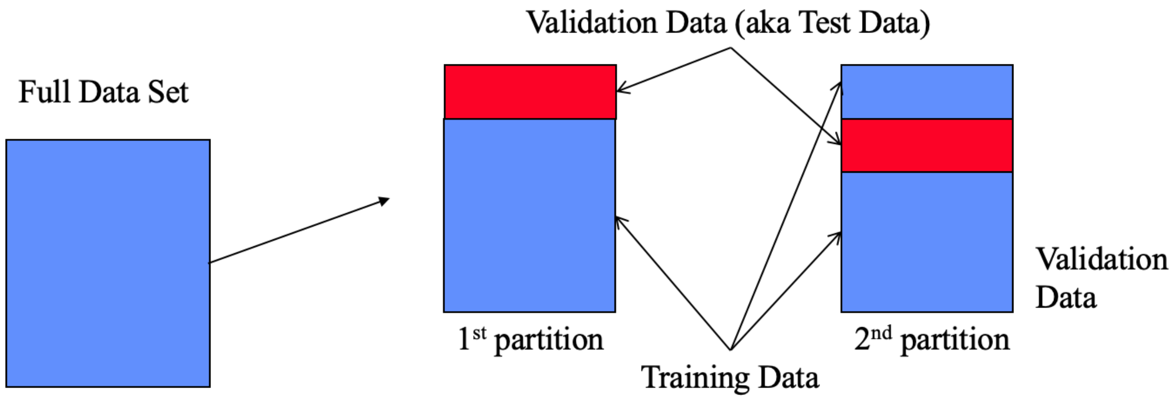
Disjoint Validation Data Sets



Methods for Performance Evaluation

- Cross-validation (*k*-fold)

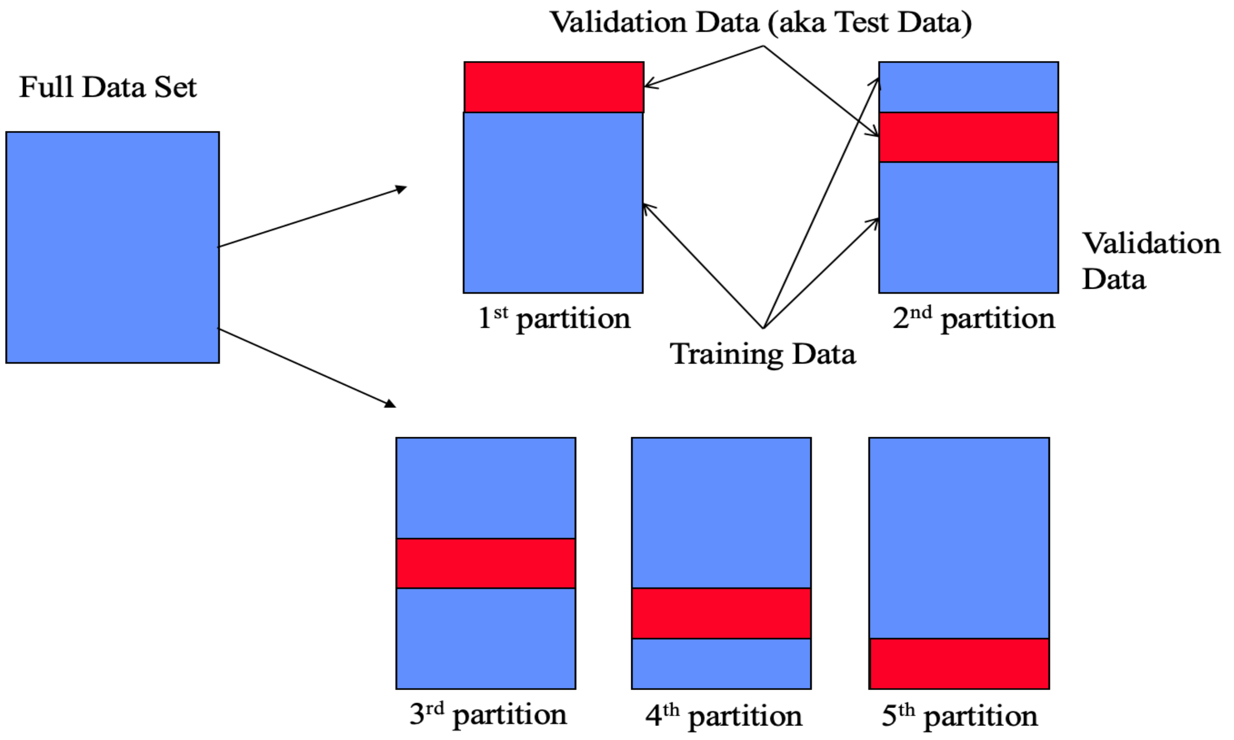
Disjoint Validation Data Sets



Methods for Performance Evaluation

- Cross-validation (k-fold)

Disjoint Validation Data Sets



Methods for Performance Evaluation

- Cross-validation (*k*-fold)

  - Notes
    - cross-validation generates an approximate estimate of how well the learned model will do on "unseen" data
    - by averaging over different partitions it is more robust than just a single train/validate partition of the data
    - "k-fold" cross-validation is a generalization
      - partition data into disjoint validation subsets of size n/k
      - train, validate, and average over the v partitions
      - e.g., k=10 is commonly used
    - k-fold cross-validation is approximately k times computationally more expensive than just fitting a model to all of the data

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Methods for Model Comparison

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TPR (on the y-axis) against FPR (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
- If the classifier returns a real-valued prediction,
  - changing the threshold of algorithm changes the location of the point

Methods for Model Comparison

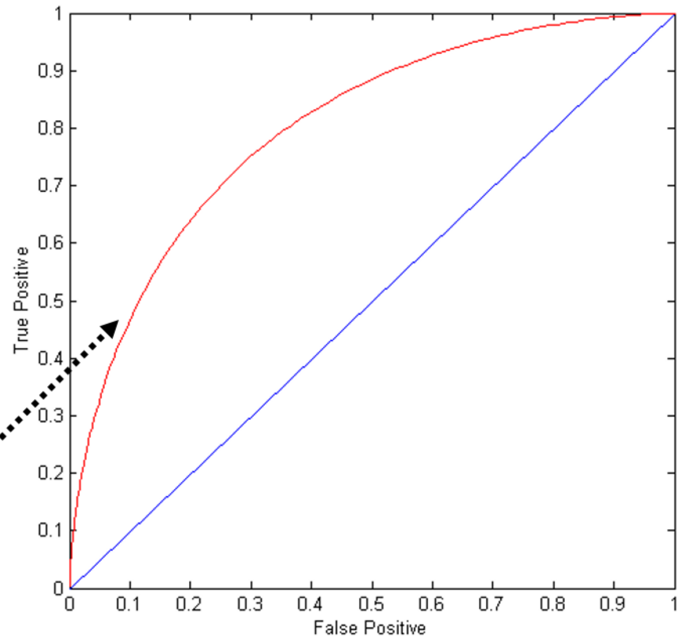
ROC (Receiver Operating Characteristic)

		PREDICTED CLASS	
		Class =Yes	Class= No
ACTUAL CLASS	Class =Yes	a (TP)	b (FN)
	Class =No	c (FP)	d (TN)

TPR = TP/(TP+FN)
FPR = FP/(FP+TN)

At threshold t:

TP=50, FN=50, FP=12, TN=88



Methods for Model Comparison

ROC (Receiver Operating Characteristic)

	PREDICTED CLASS	
	Class =Yes	Class= No
ACTUAL CLASS Class =Yes	a (TP)	b (FN)
ACTUAL CLASS Class =No	c (FP)	d (TN)

TPR = TP/(TP+FN)
FPR = FP/(FP+TN)

(TPR,FPR):

- (0,0): declare everything to be negative class
  - TP=0, FP = 0
- (1,1): declare everything to be positive class
  - FN = 0, TN = 0
- (1,0): ideal
  - FN = 0, FP = 0

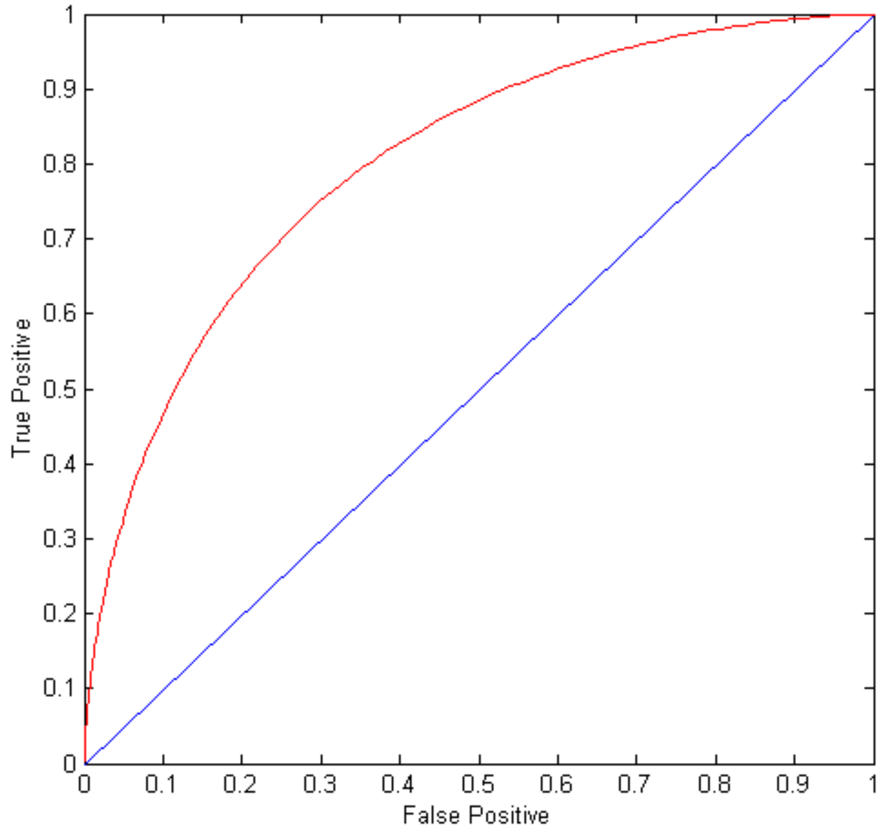
Methods for Model Comparison

ROC (Receiver Operating Characteristic)

(TPR,FPR):

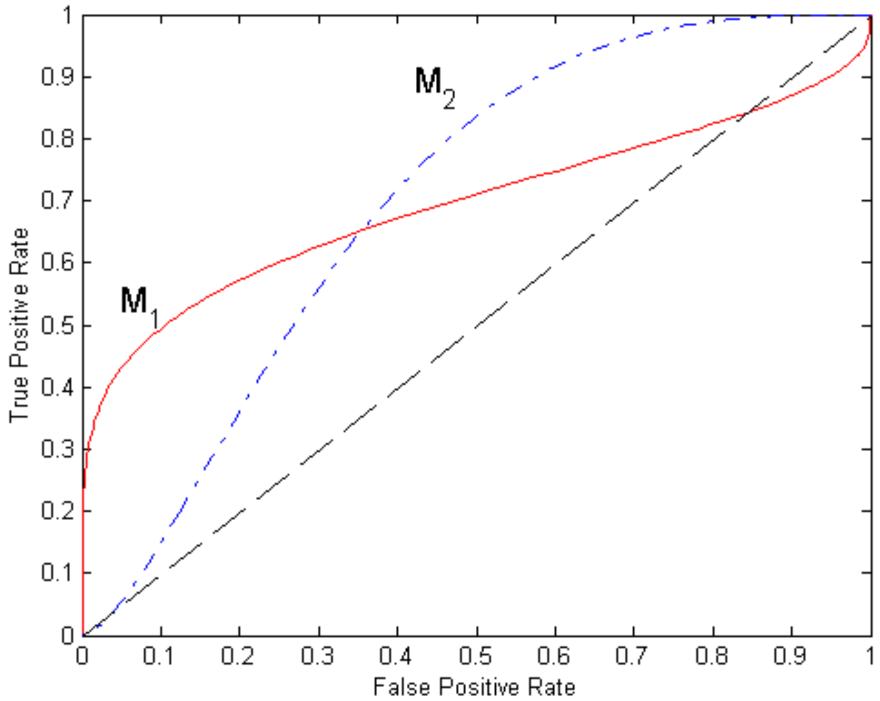
- (0,0): declare everything
  to be negative class
- (1,1): declare everything
  to be positive class
- (1,0): ideal

- Diagonal line:
  - Random guessing
  - Below diagonal line:
    - prediction is opposite of the true class



Methods for Model Comparison

Using ROC for Model Comparison



- No model consistently outperforms the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR
- Area Under the ROC curve (AUC)
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5