Business Intelligence

TICS-423
Universidad Adolfo Ibáñez

Week 07: 12 September - 16 September, 2016

Claudio Diaz
Sebastián Moreno
Gonzalo Ruz

Predictive modelling
Model evaluation

Predictive modeling

- **Task Specification: Predictive Modeling**
- **Data Representation: Homogeneous IID data**
- **Knowledge representation:**
- **Learning technique**
  - **Search + Scoring**
- **Prediction and/or interpretation**

Predictive modeling, model evaluation

- **Overfitting**
  What is overfitting?

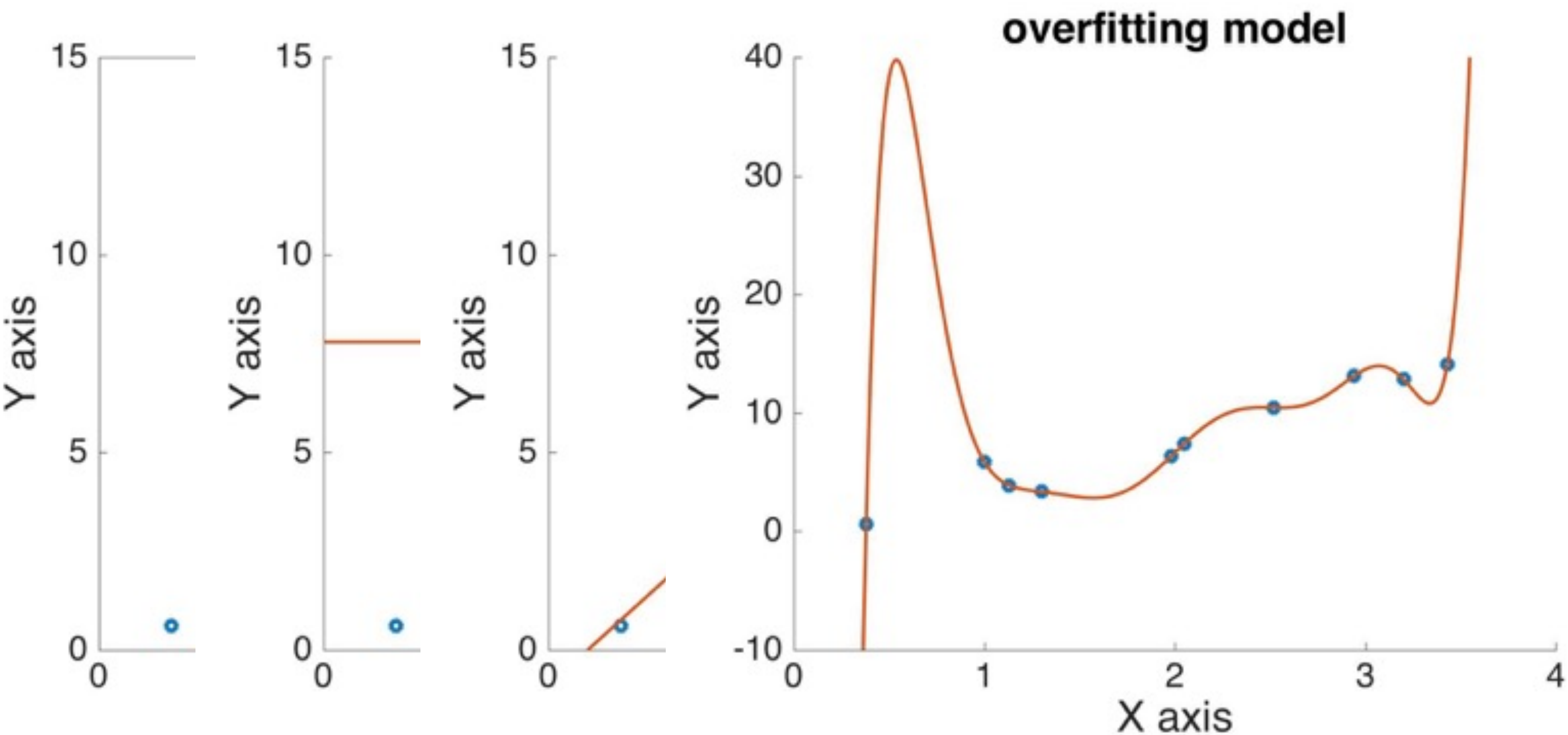
- Metrics for Performance Evaluation
  How to evaluate the performance of a model?

- Methods for Performance Evaluation
  How to obtain reliable estimates?

- Methods for Model Comparison
  How to compare the relative performance among competing models?

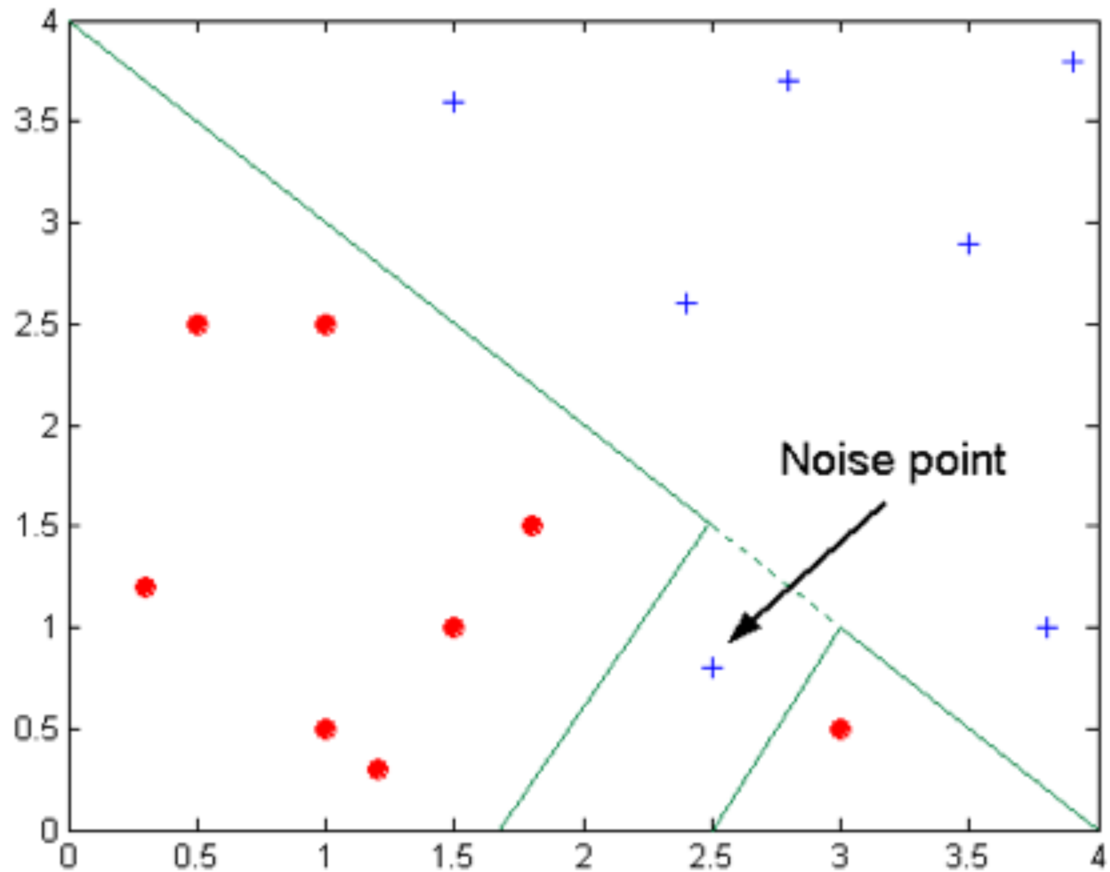
Predictive modeling, model evaluation, overfitting

- In overfitting, a statistical model describes random error or noise instead of the underlying relationship. Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.



Predictive modeling, model evaluation, overfitting

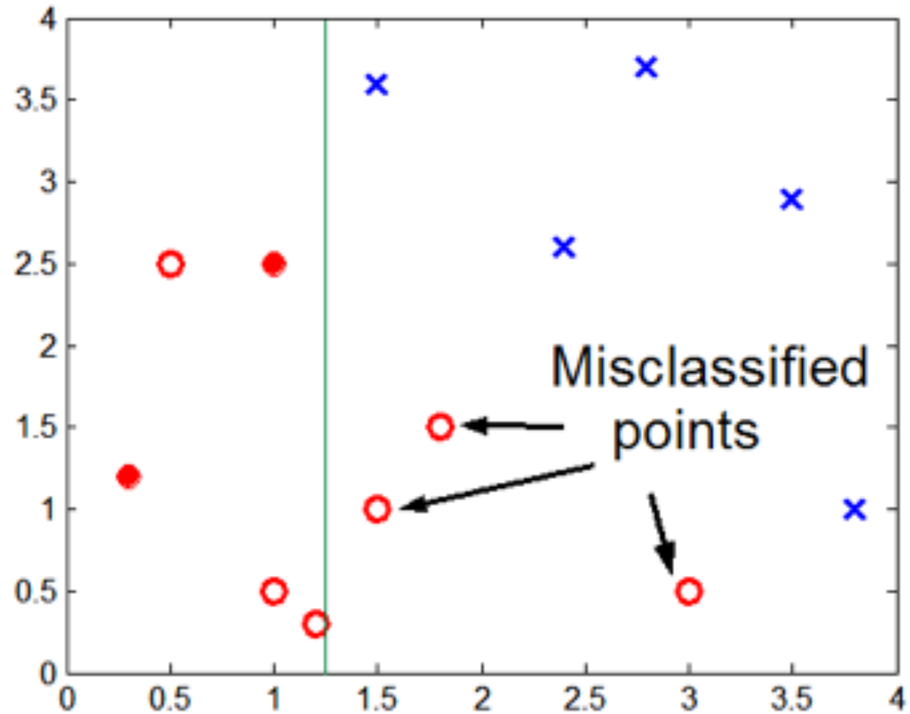
- In overfitting, a statistical model describes random error or noise instead of the underlying relationship.



Decision boundary is distorted by noise point

Predictive modeling, model evaluation, overfitting

- In overfitting, a statistical model describes random error or noise instead of the underlying relationship.



- Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region.

Predictive modeling, model evaluation

- Overfitting
  What is overfitting?

- **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?

PM, model evaluation, performance evaluation

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

<table>
<tr><th colspan="2" rowspan="2">Confusion matrix</th><th colspan="2">Predicted Class</th></tr>
<tr><th>No</th><th>Yes</th></tr>
<tr><th rowspan="2">Actual Class</th><th>No</th><td>True Negative</td><td>False Positive</td></tr>
<tr><th>Yes</th><td>False Negative</td><td>True Positive</td></tr>
</table>

- Associated metrics for confusion matrix:

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{F1-score} = \frac{2 * TP}{2 * TP + FP + FN}$$

PM, model evaluation, performance evaluation

- Accuracy could mislead in biased data.
- Recall is biased towards the positive data.
- Precision is biased towards **predicted** positive data.
- F1-score is biased towards all data except true negative
- Example: 2 class problem
  Number of Class 0 examples = 9900
  Number of Class 1 examples = 100

Confusion matrix		**Predicted Class** 0	**Predicted Class** 1
Actual Class	0	9760	140
Actual Class	1	40	60

- Accuracy = 9820/10000=98.2%
- Recall = 60/100=60.0%
- Precision = 60/200 = 30.0%
- F1-score = 120/300 =40.0%

PM, model evaluation, performance evaluation

- **Cost matrix:** In some problems there are costs associated with a wrong or correct classification.

Confusion matrix		Predicted Class	
		No	Yes
Actual Class	No	True Negative	False Positive
	Yes	False Negative	True Positive

Cost Matrix		Predicted Class	
		No	Yes
Actual Class	No	C(No|No)	C(Yes|No)
	Yes	C(No|Yes)	C(Yes|Yes)

C(i|j): Cost of misclassifying class j example as class i

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
  Recall is biased towards C(Yes|Yes) & C(No|Yes)
  F1-score is biased towards all except C(No|No)

PM, model evaluation, performance evaluation

Cost matrix		Predicted Class	
	C(i|j)	+	-
Actual Class	+	-1	100
	-	1	0

Model 1		Predicted Class	
		+	-
Actual Class	+	150	40
	-	60	250

Model 2		Predicted Class	
		+	-
Actual Class	+	250	45
	-	5	200

Accuracy = 80%
Cost = 3910

Accuracy = 90%
Cost = 4255

PM, model evaluation, performance evaluation

- Performance evaluation for predictive models:

$$S(M) = \sum_{i=1}^{N_{test}} d[f(x(i); M), y(i)]$$

Sum over examples

Distance between predicted and true

Predicted class label for item *i*

True class label for item *i*

PM, model evaluation, performance evaluation

- Common performance evaluations:
  - Zero-one loss: $$S_{0/1}(M) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} I[f(x(i); M), y(i)]$$
    $$\text{where } I(a,b) = \begin{cases} 1 & a \neq b \\ 0 & \text{otherwise} \end{cases}$$
  - Squared loss: $$S_{sq}(M) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} [f(x(i); M) - y(i)]^2$$

Predictive modeling, model evaluation

- Overfitting
  What is overfitting?

- 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?

Predictive modeling, model evaluation, methods

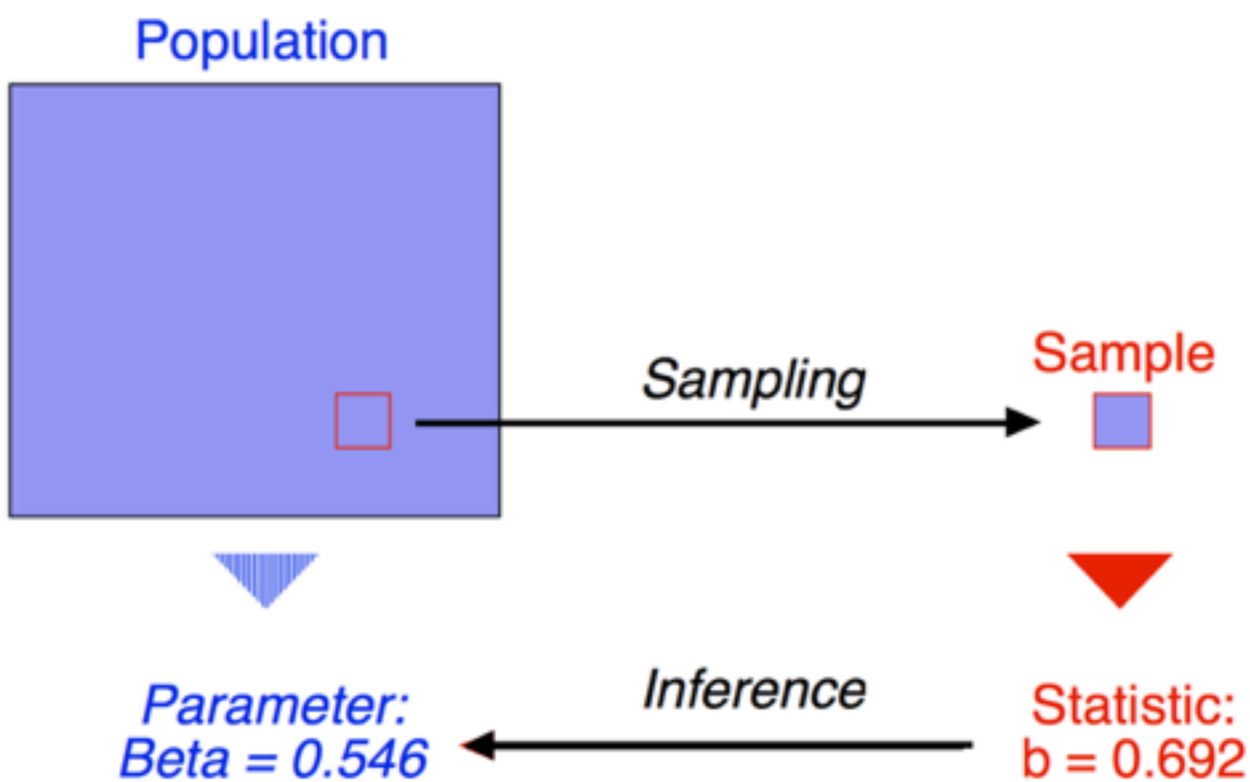
- Our goal is to estimate true future error rate using the current **sample** of the data set
- Approaches:
    - Reclassify training data to estimate error rate
    - Classify disjoint test set to estimate generalization rate
        - Disjoint subsets
        - Overlapping subsets
    - Cross validation

PM, model evaluation, methods, sampling

- In data mining we often work with a sample of data from the population of interest.

- **Estimation** techniques allow inferences about population properties from sample data.

- If we had the population we could calculate the properties of interest.



PM, model evaluation, methods, sampling

- Elementary units:
  - Entities (e.g., persons, objects, events) that meet a set of specified criteria
  - Example: All people who’ve purchased something from Walmart in the past month
- Population:
  - Aggregate of elementary units (i.e, all items of interest)
- Sampling:
  - Sub-group of the population
  - Serves as a reference group for estimating characteristics about the population and drawing conclusions

PM, model evaluation, methods, sampling

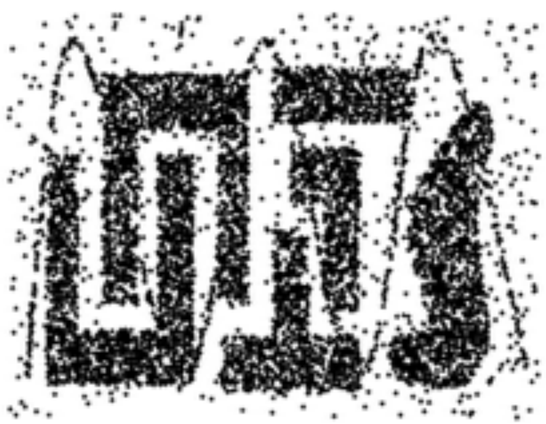
- Sampling is the main technique employed for data selection: It is often used for both the preliminary investigation of the data and the final data analysis
- **Reasons** to sample
  - Obtaining/processing the entire set of data of interest is too expensive or time consuming
  - Note: even if you use an entire dataset for analysis, you should be aware of the sampling method that was used to gather the dataset
- The **key principle** for effective sampling is the following:
  - Using a sample will work almost as well as using the entire data set, if the sample is **representative**
  - A sample is representative if it has approximately the same property (of interest) as the original set of data

PM, model evaluation, methods, sampling

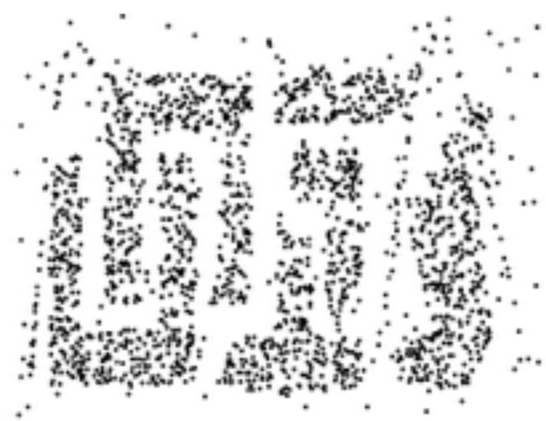
- Types of probability sampling:
- **Simple random sampling:** There is an equal probability of selecting any particular item
- **Sampling without replacement:** As each item is selected, it is removed from the population
- **Sampling with replacement:** Items are not removed from the population as they are selected for the sample; the same item can be picked up more than once
- **Stratified sampling:** Split the data into several partitions; then draw random samples from each partition

PM, model evaluation, methods, sampling

- How does sample size affect learning?



8000 points

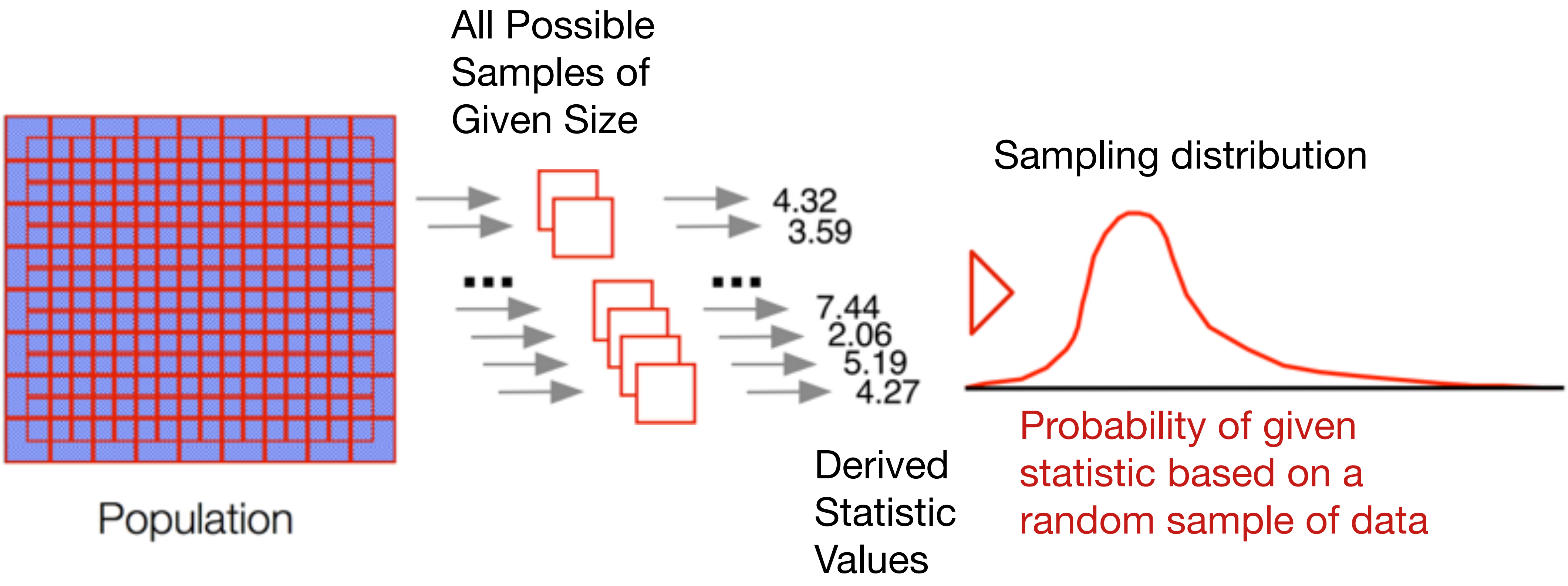


2000 Points

500 Points

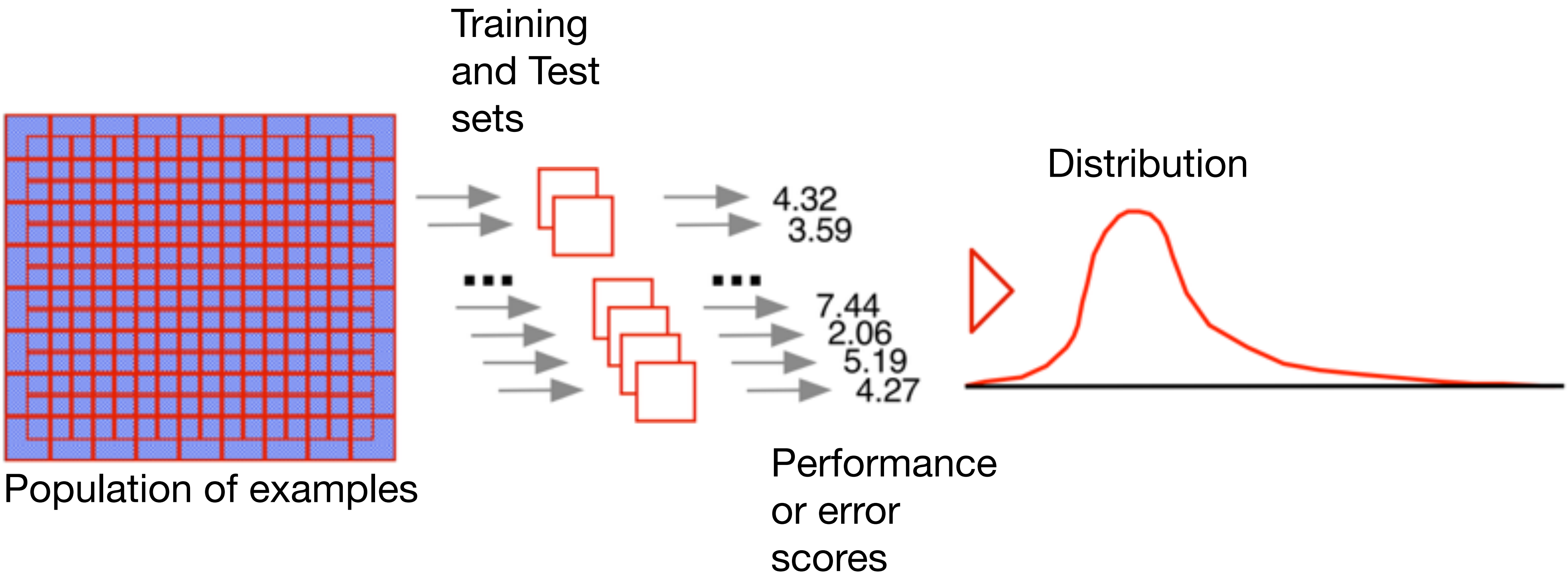
PM, model evaluation, methods, sampling

- Sampling distributions



PM, model evaluation, methods, sampling

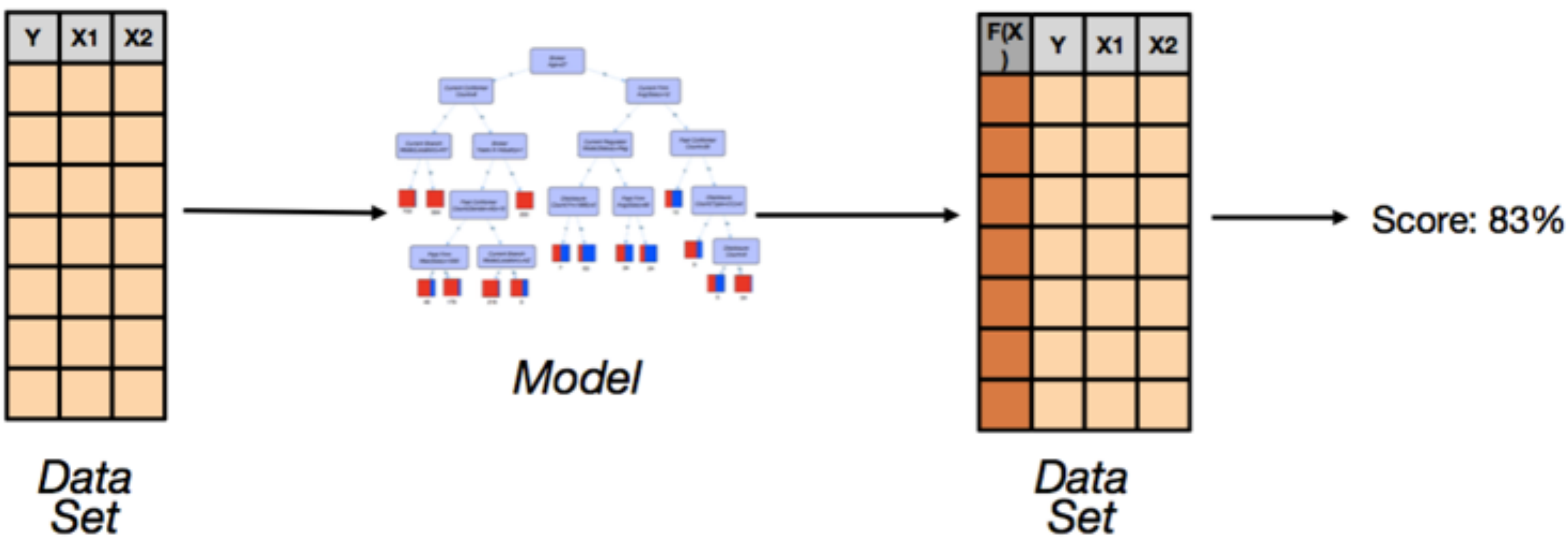
- To estimate the error of a model we can also use sampling to estimate the distribution of the error.



- Distribution => there is a mean and variance of the error distribution.

PM, model evaluation, methods, reclassify

- Reclassify training data to estimate error rate

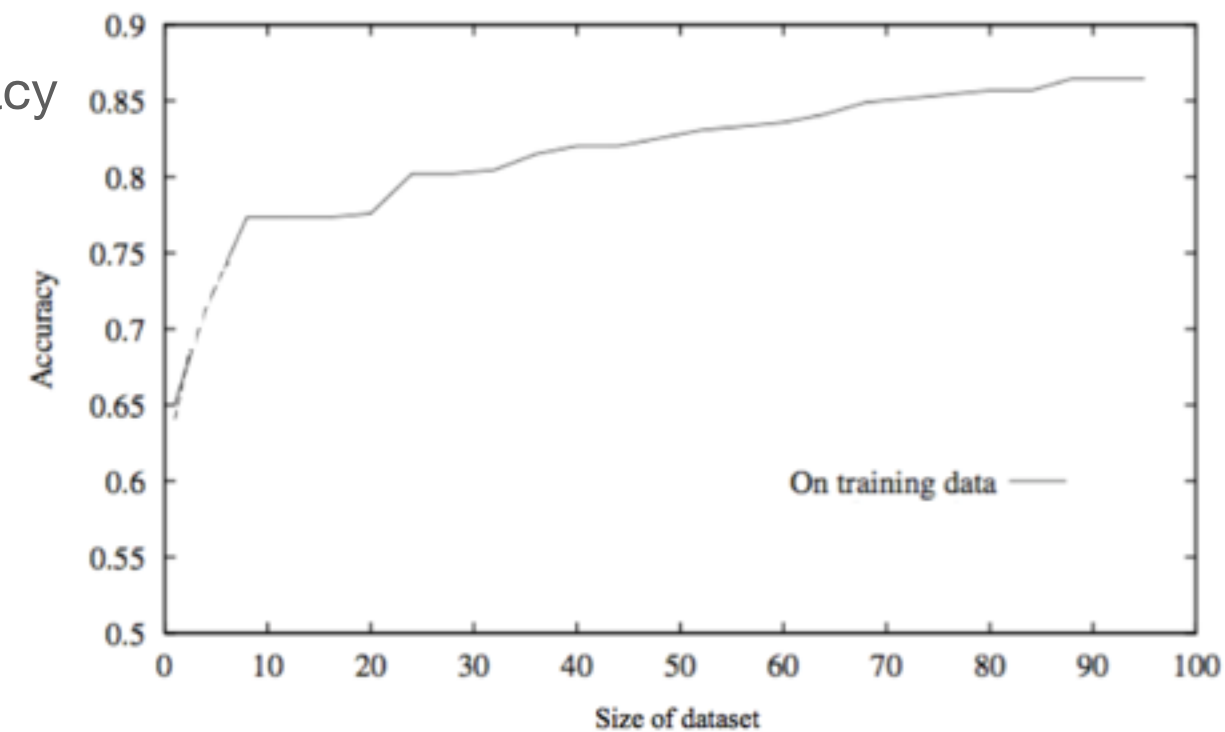


- Estimates a single point of the future error instead of a distribution, and typically, this estimation is biased.

PM, model evaluation, methods, reclassify

- Learning curve: it shows how accuracy changes with varying sample size.

- From dataset set S, where |S|=n
  For i=[10, 20, ... ,100]
  Randomly sample i% of S to construct sample S'
  Learn model on S'
  Evaluate model on S
  Plot training set size vs. accuracy

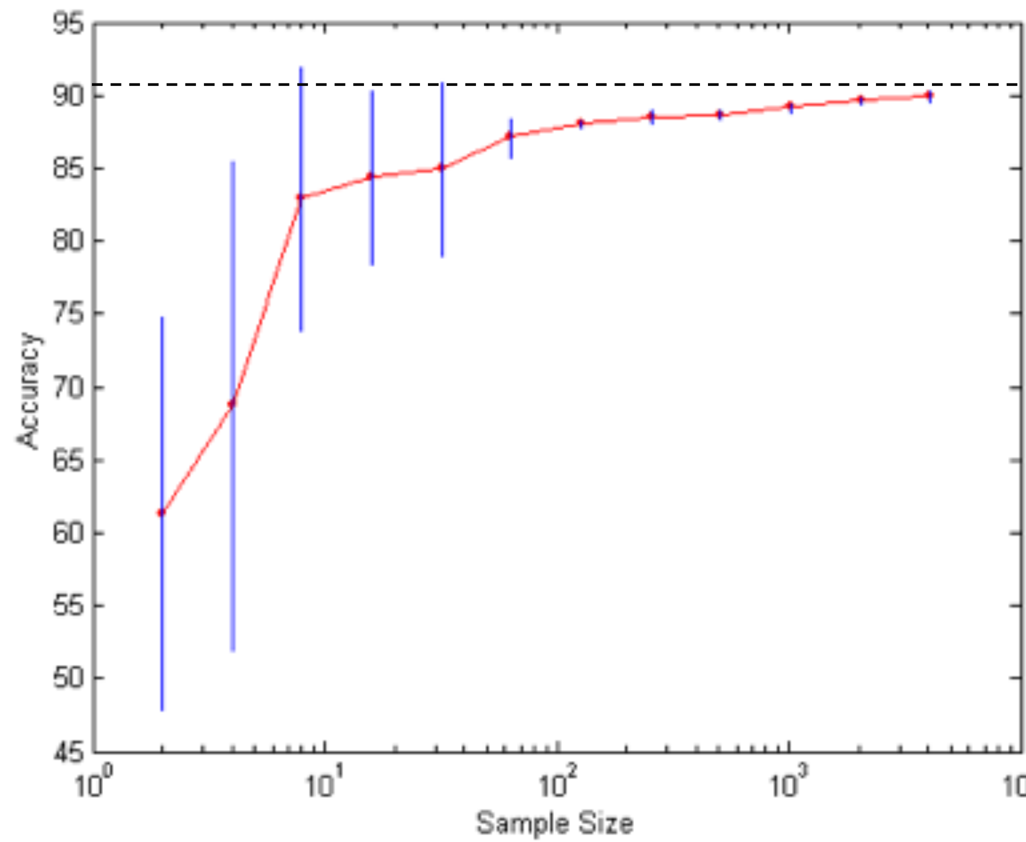


- Effect of small sample size:
  Bias in the estimate
  Variance of estimate

PM, model evaluation, methods, reclassify

- Learning curve: it shows how accuracy changes with varying sample size.

- From dataset set S, where |S|=n
  For i=[10, 20, ... ,100]
  Randomly sample i% of S to construct sample S'
  Learn model on S'
  Evaluate model on S
  Plot training set size vs. accuracy

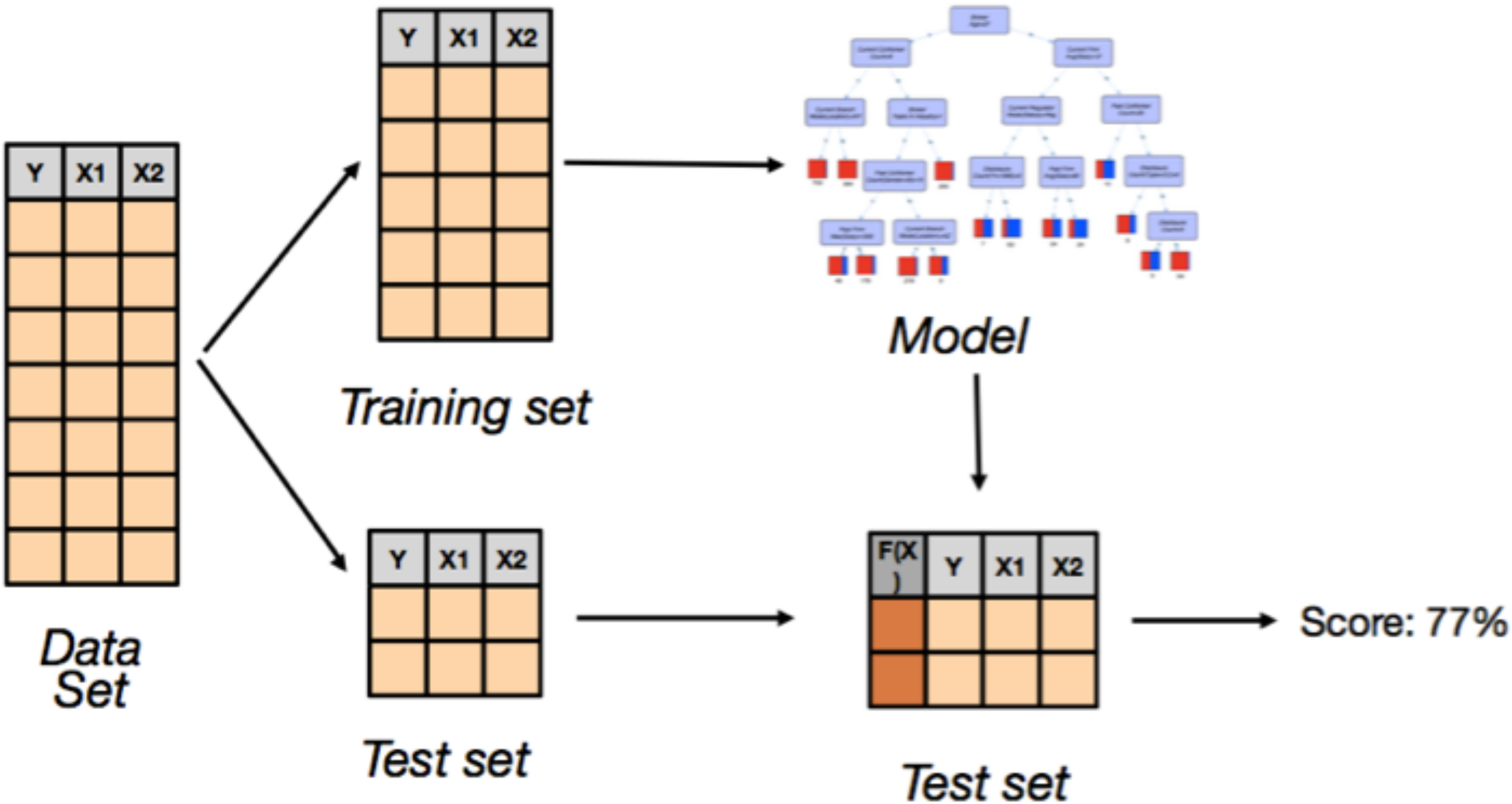


- Effect of small sample size:
  Bias in the estimate
  Variance of estimate

- To calculate the standard deviation repeat the process several times

PM, model evaluation, methods, disjoint

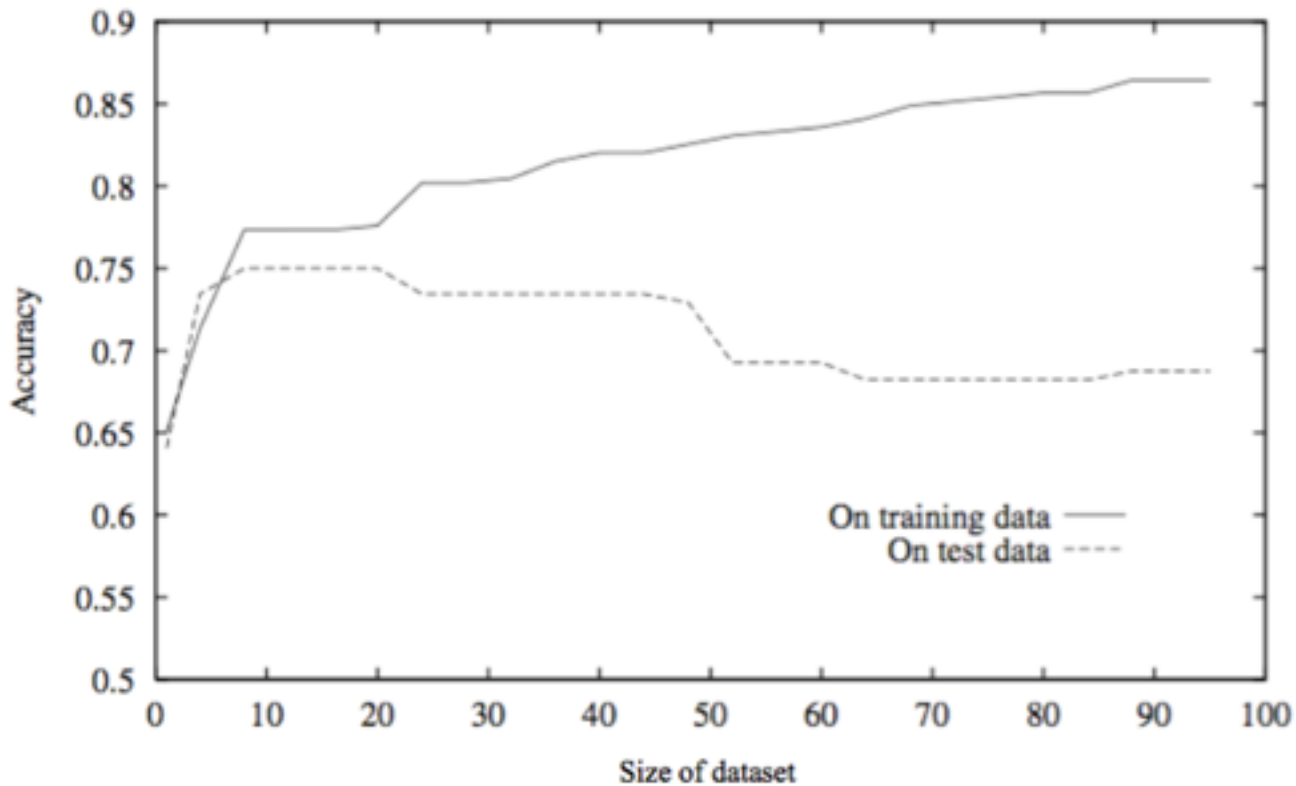
- Classify **disjoint** test set to estimate generalization rate



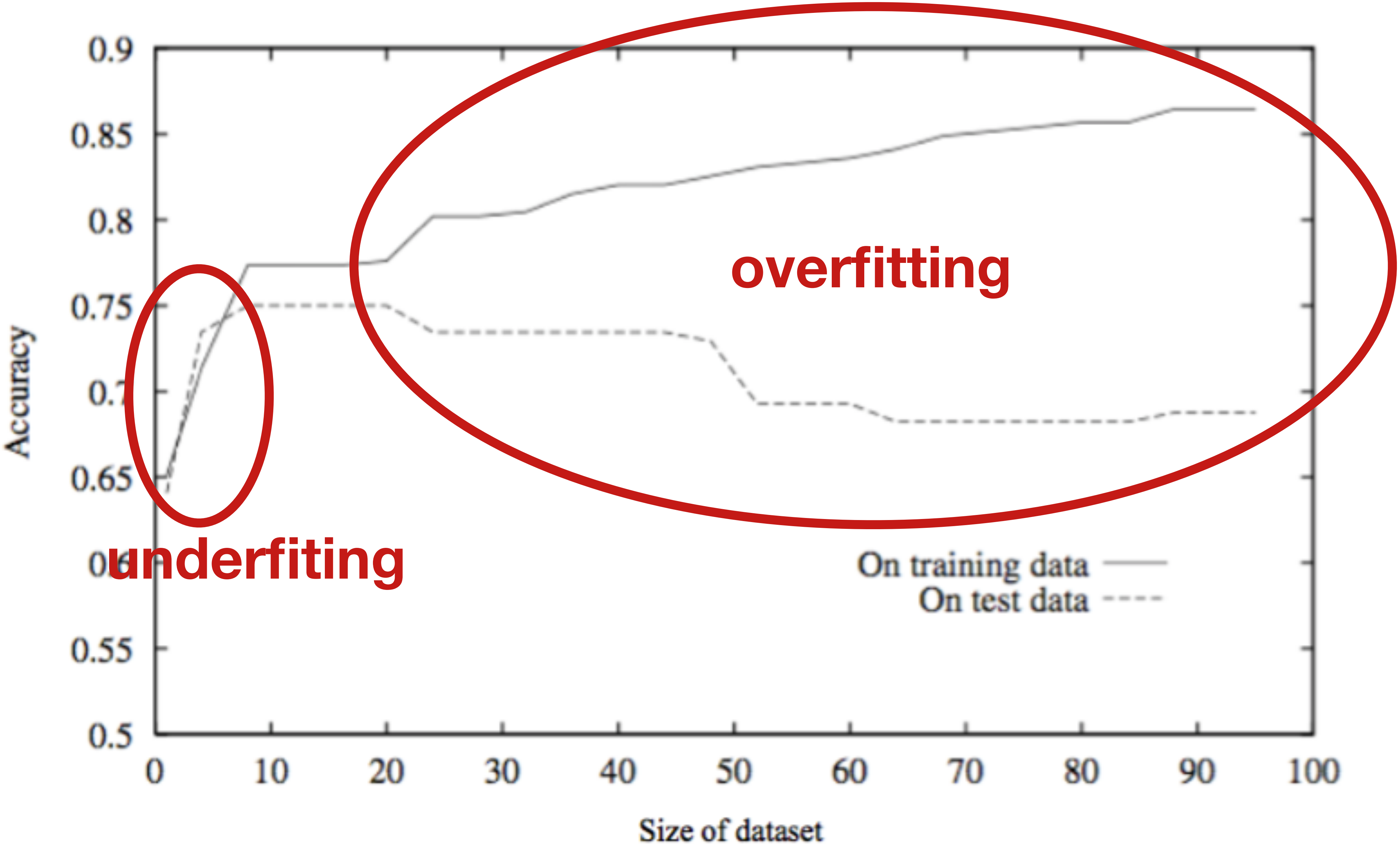
- Estimate will vary due to size and makeup of test set

PM, model evaluation, methods, disjoint

- From dataset set S, split the data set in S_{train} and S_{test}
  For i=[10, 20, ... ,100]
  Randomly sample i% of S_{train} to construct sample S’
  Learn model on S’
  Evaluate model on S_{test}
  Plot training set size vs. accuracy



PM, model evaluation, methods, disjoint



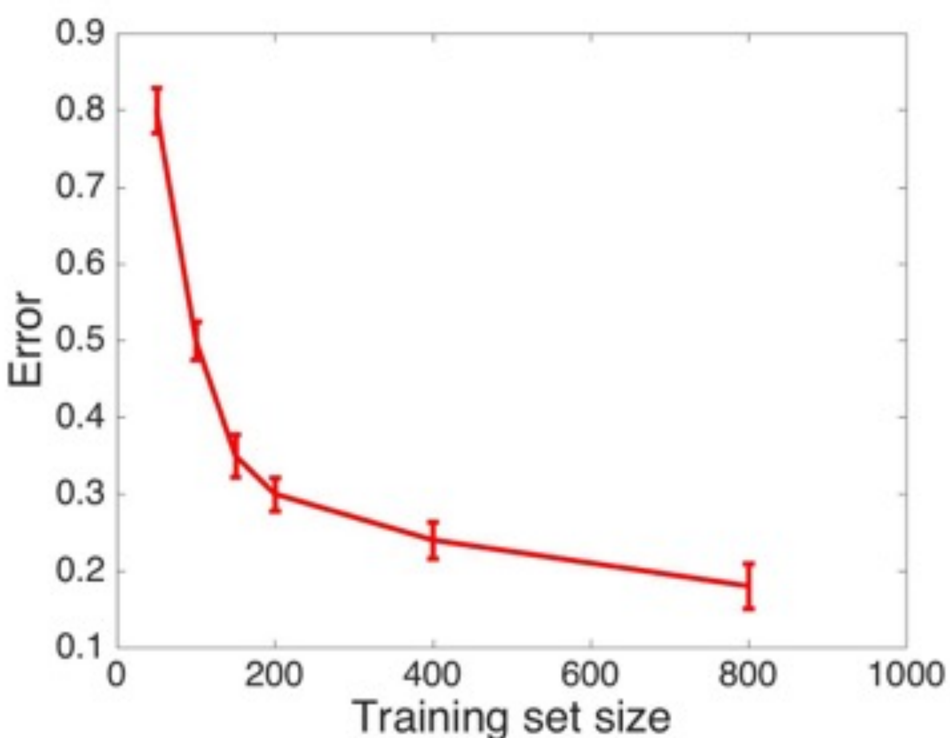
PM, model evaluation, methods, disjoint

- To calculate the standard deviation of the model error, repeat the disjoint process of the data multiple times and calculate the average.

- For k=1 to k (defined by the user)
  Split data S set in S_{train} and S_{test}
  For i=[10, 20, ... ,100]
    Randomly sample i% of S_{train} to construct sample S'
    Learn model on S'
    Evaluate model on S_{test}
  Average error rates over the k trials
  Plot average error and standard deviation

- Repeated sampling of test sets leads to overlap (i.e., dependence) among test sets; resulting in underestimation of variance

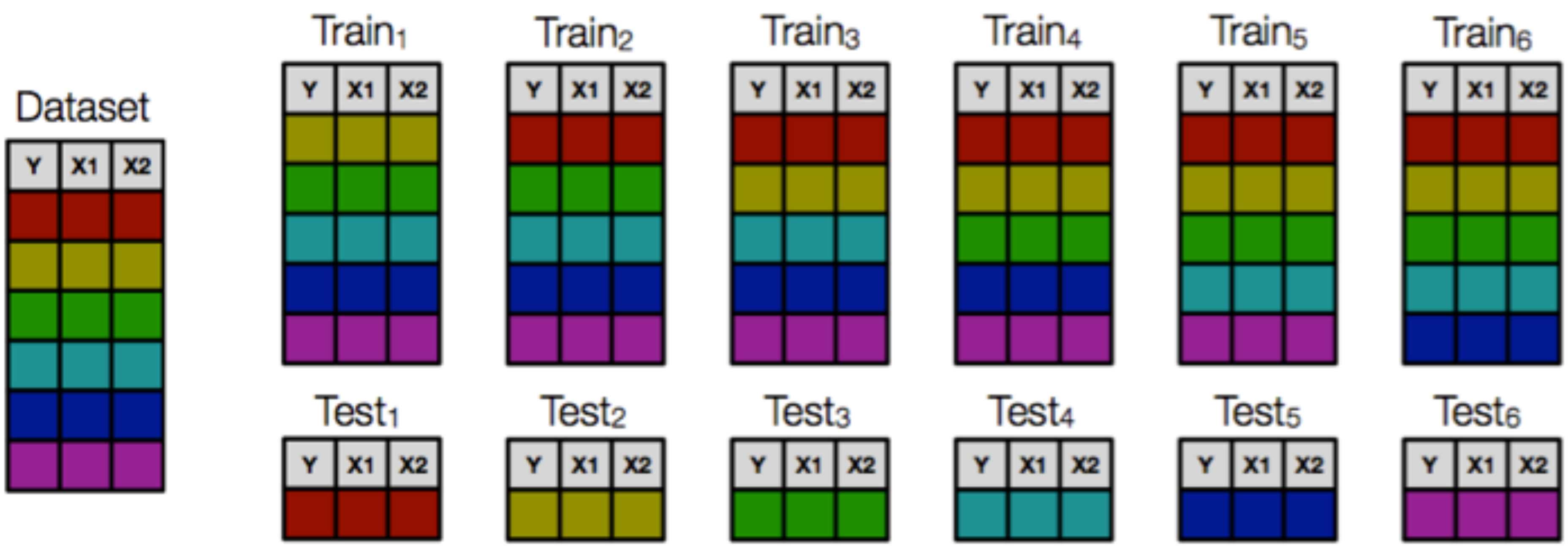
- Standard errors will be **biased** if performance is estimated from **overlapping** test sets (Dietterich'98)



PM, model evaluation, methods, cross-validation

- K-fold cross validation combines (averages) measures of fit (prediction error) to derive a more accurate estimate of model prediction performance

- Randomly **partition** training data into k folds
  For i=1 to k
  Learn model on D - ith fold;
  evaluate model on ith fold
  Average results from all k trials

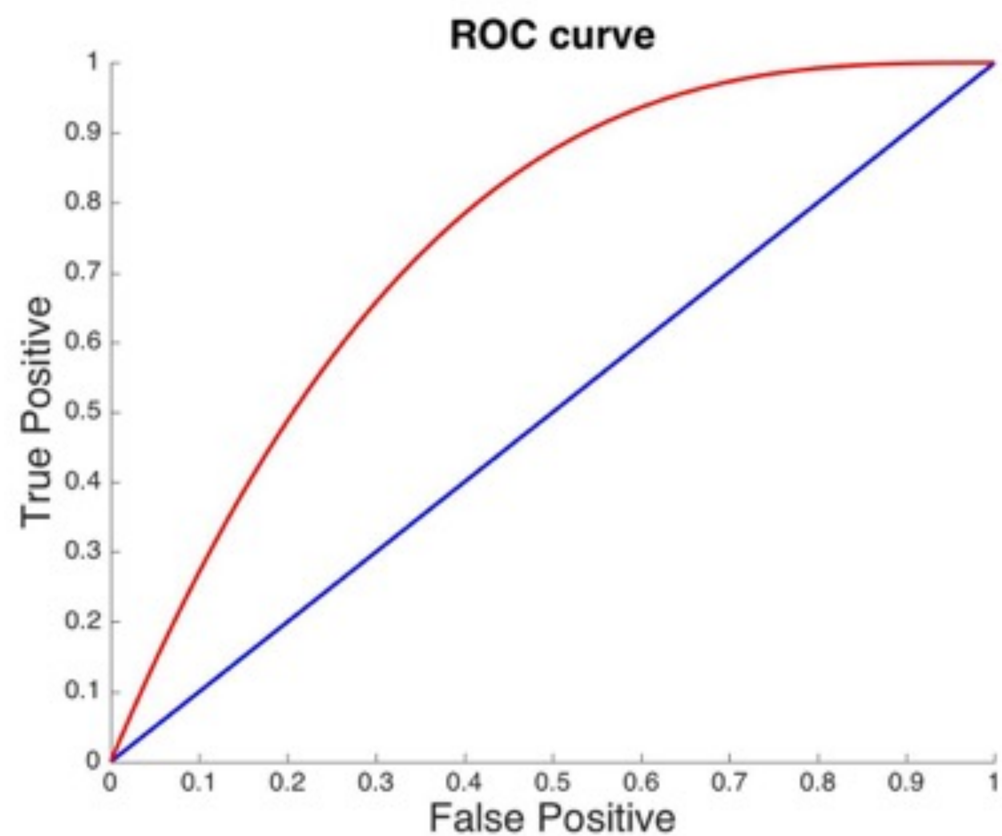


PM, model evaluation, methods, cross-validation

- K-fold cross validation can be used is several cases
- Parameter setting
  - Decision tree example: Choose threshold for split function with cv
    - Repeatedly learn model with different thresholds
    - Pick threshold that shows best cross-validation performance
- Model evaluation
  - Estimate model performance across k-fold cv trials
  - Use performance measurement as empirical sampling distribution for model performance

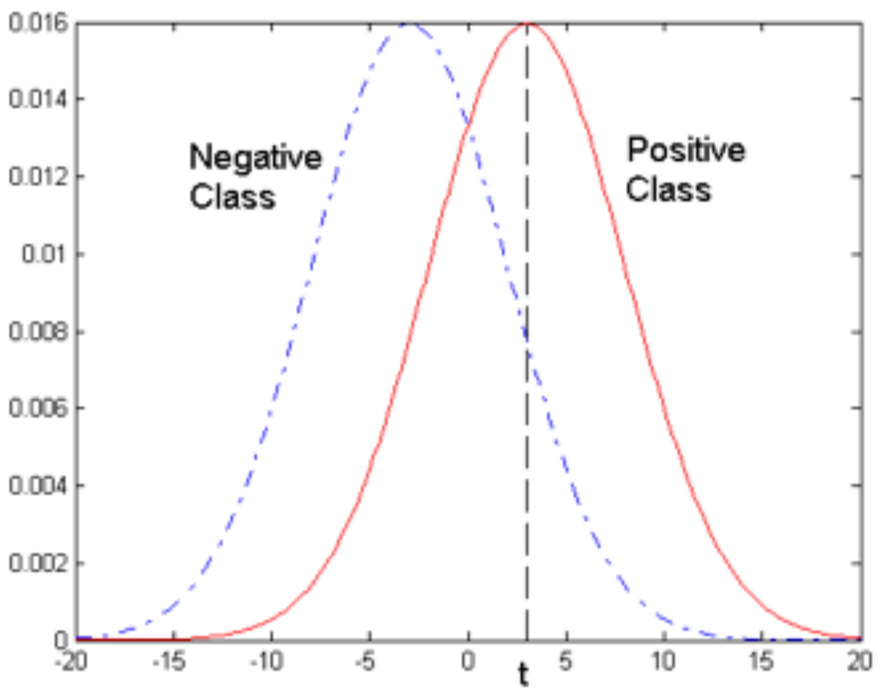
PM, model evaluation, methods, ROC curve

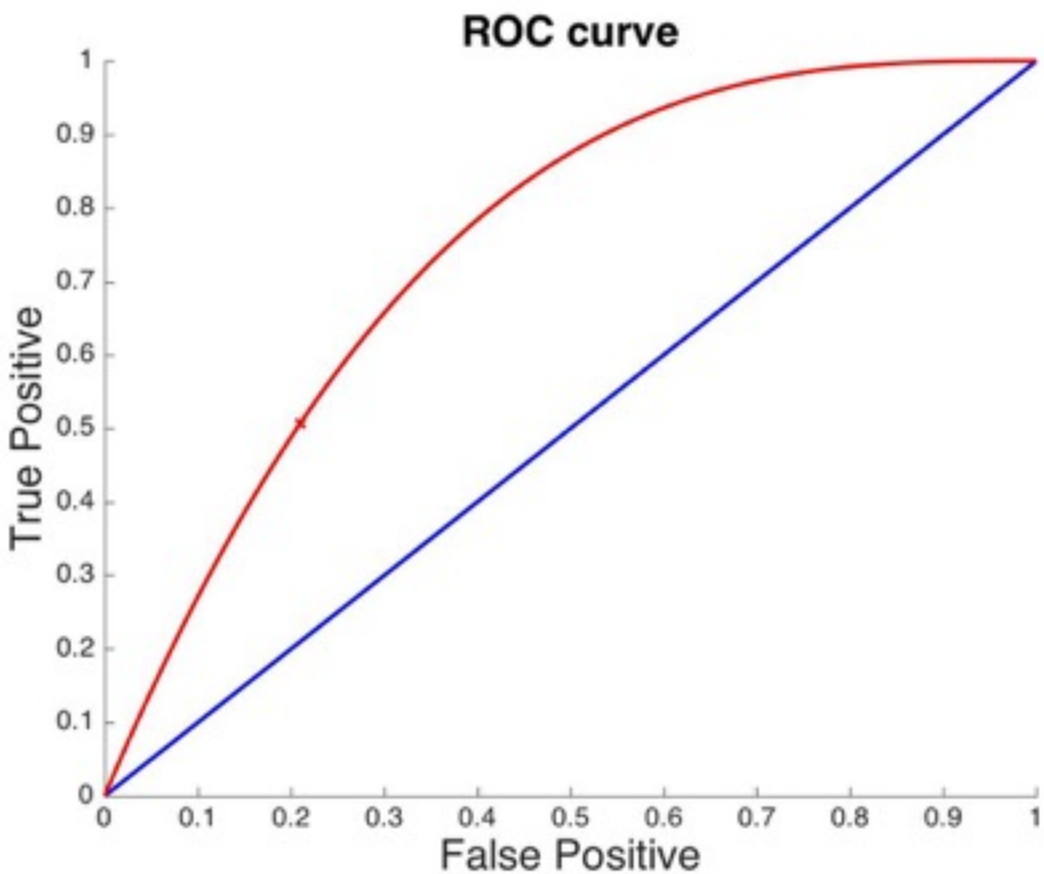
- Developed in 1950s for signal detection theory to analyze noisy signals Characterize the trade-off between positive hits and false alarms
- ROC curve plots True Positive rate (TP) on the y-axis against False Positive rate on the x-axis.
- Performance of each classifier represented as a point on the ROC curve. Changing the threshold of algorithm, sample distribution or cost matrix, changes the location of the point, which generates the final curve.
- Area Under the Curve (AUC): is the area below the ROC curve, and summarise the performance of the model.



PM, model evaluation, methods, ROC curve

- 1-dimensional data set containing 2 classes (positive and negative). Any points located at x > t is classified as positive





- At threshold t TP=0.5 and FP=0.21

PM, model evaluation, comparison, ROC curve

- To construct a ROC curve:
- Use classifier that produces posterior probability for each test instance P(+|A)
- Sort the instances according to P(+|A) in decreasing order
- Apply threshold at each unique value of P(+|A)
- Count the number of TP, FP, TN, FN at each threshold and calculate
  TP rate, TPR = TP/(TP+FN)
  FP rate, FPR = FP/(FP + TN)

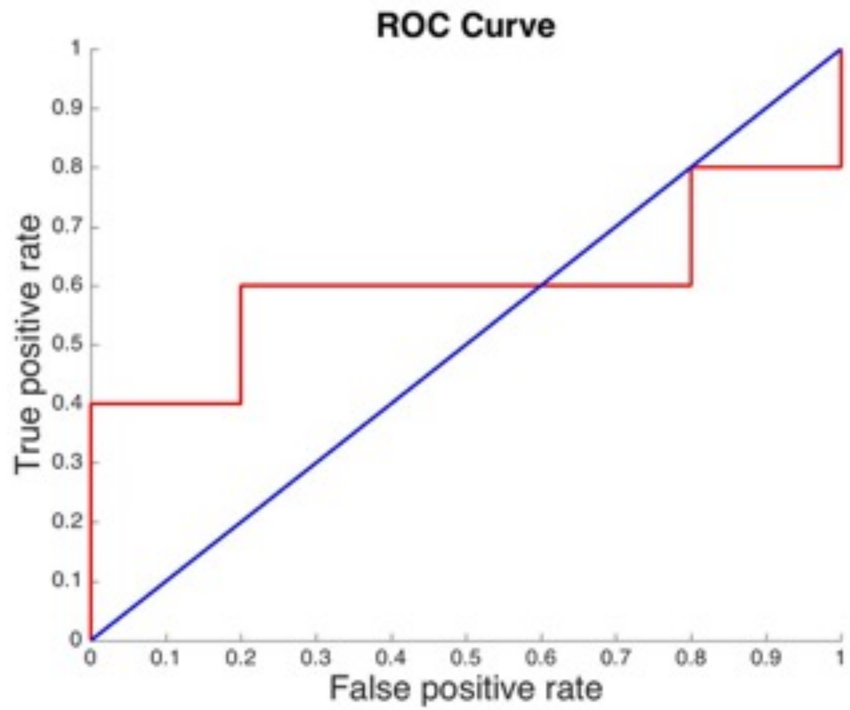
Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

PM, model evaluation, comparison, ROC curve

- Example

Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0



Predictive modeling, model evaluation

- Overfitting
  What is overfitting?

- Metrics for Performance Evaluation
  How to evaluate the performance of a model?

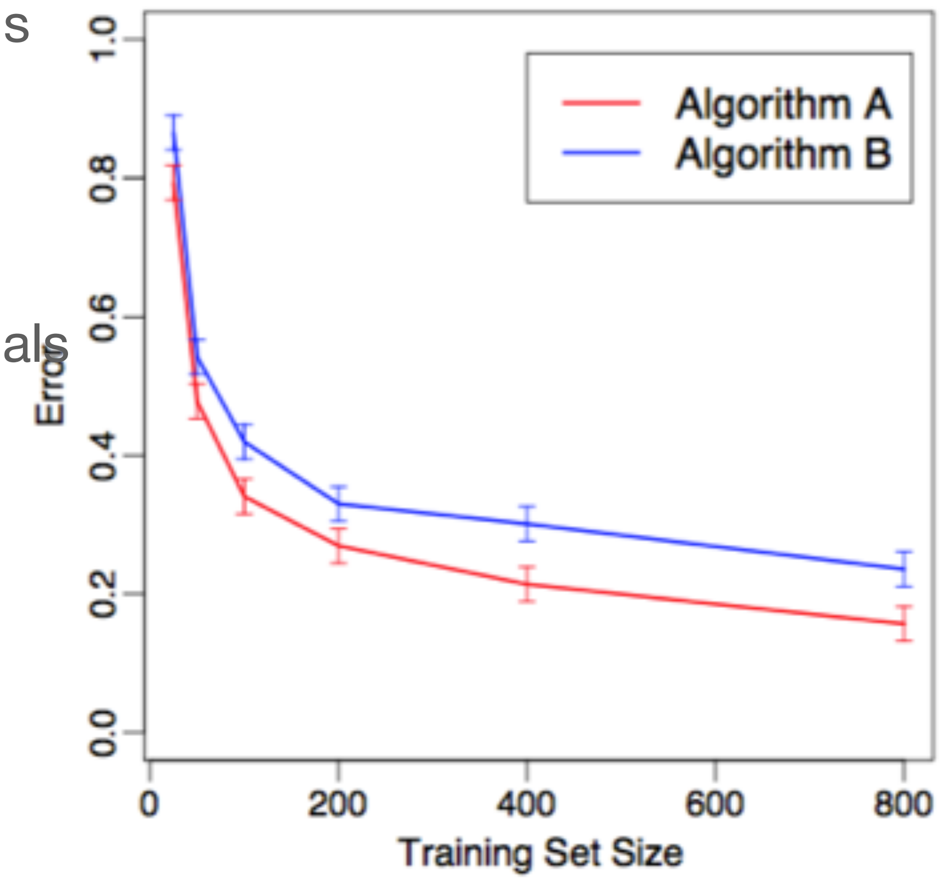
- Methods for Performance Evaluation
  How to obtain reliable estimates?

- **Methods for Model Comparison**
  How to compare the relative performance among competing models?

PM, model evaluation, comparison, cross validation

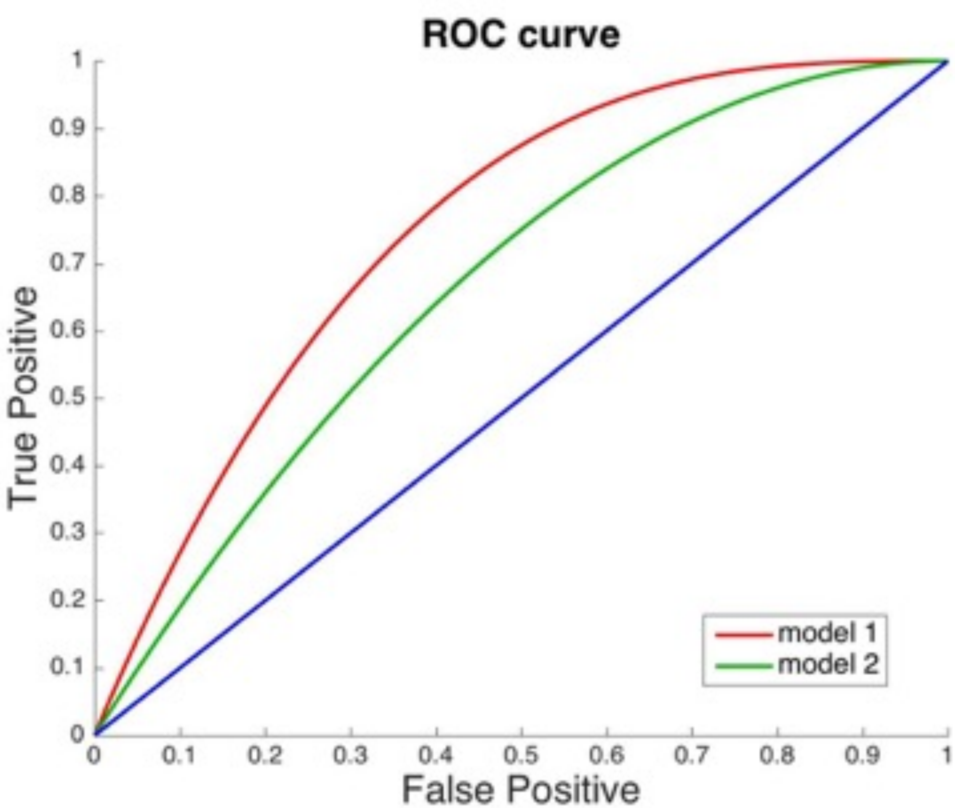
- To compare among models performance using K-fold cross validation.

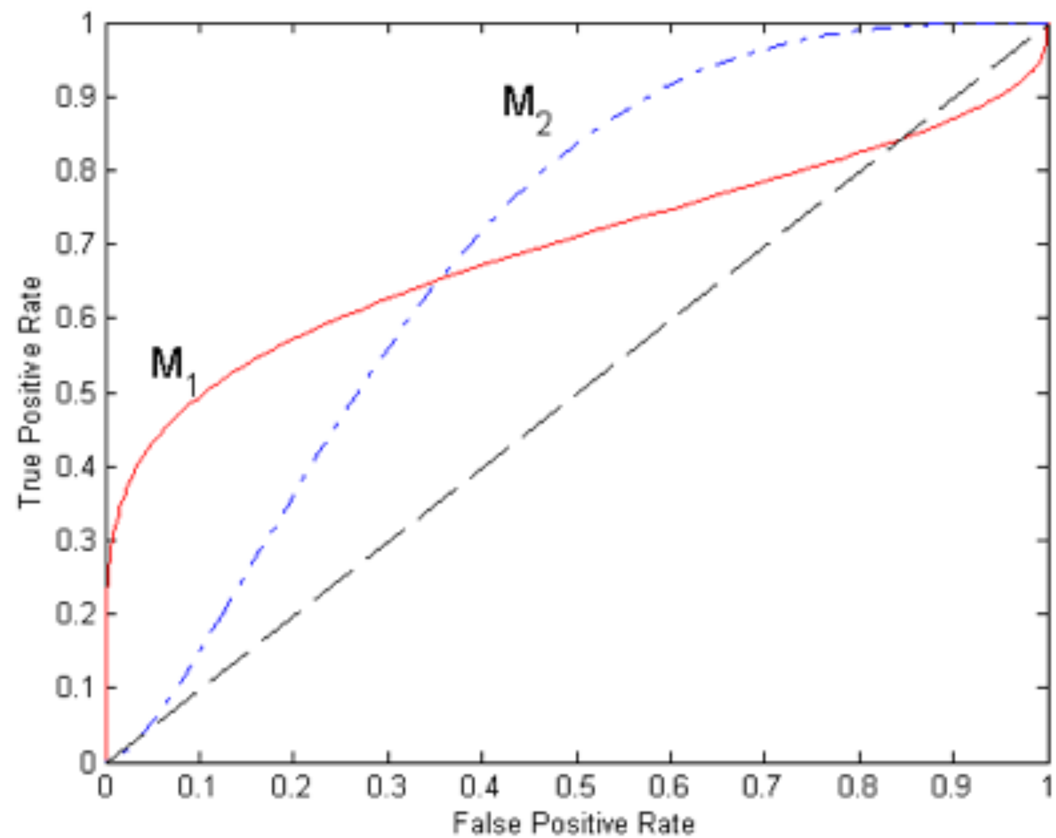
- Randomly partition training data into k folds
  for j=1 to m
      for i=1 to k
          learn model j on D - ith fold;
          evaluate model j on ith fold
      average results for model j from all k trials
  plot error with standard deviation
  compare models



PM, model evaluation, comparison, ROC curve

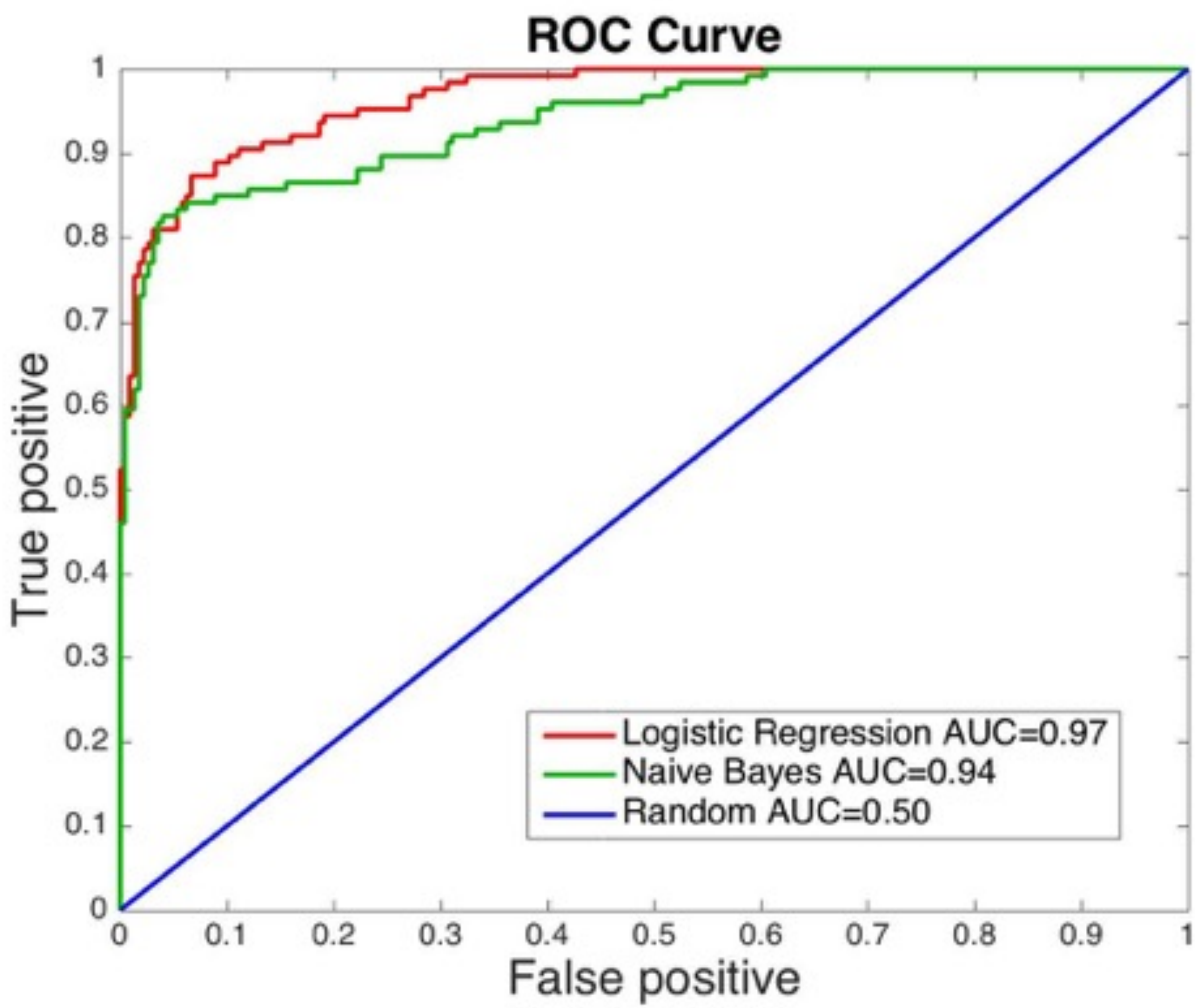
- We can visually compare models using ROC curve





PM, model evaluation, comparison, ROC curve

- We can also compare the model based on the Area Under the Curve (AUC)



PM, model evaluation, comparison, AIC y BIC

- **Occam razor**: Given two models with similar error, one must pick the simplest model instead of the complex model.

- Complex models have higher probability to model data by chance.

- **Akaike Information Criterion (AIC)**: it estimates the information lost of a model representing its data generation process.

$$\underbrace{AIC(M) = 2\ln(L(\mathbf{X}, M))}_{\text{Likelihood of the model}} - 2\underbrace{\#(M)}_{\text{number of parameter of the model}}$$

- **Bayesian Information Criteria (BIC)**: it estimates the information lost of a model representing its data generation process. It weights the penalization factor (K), by the number of points of the data.

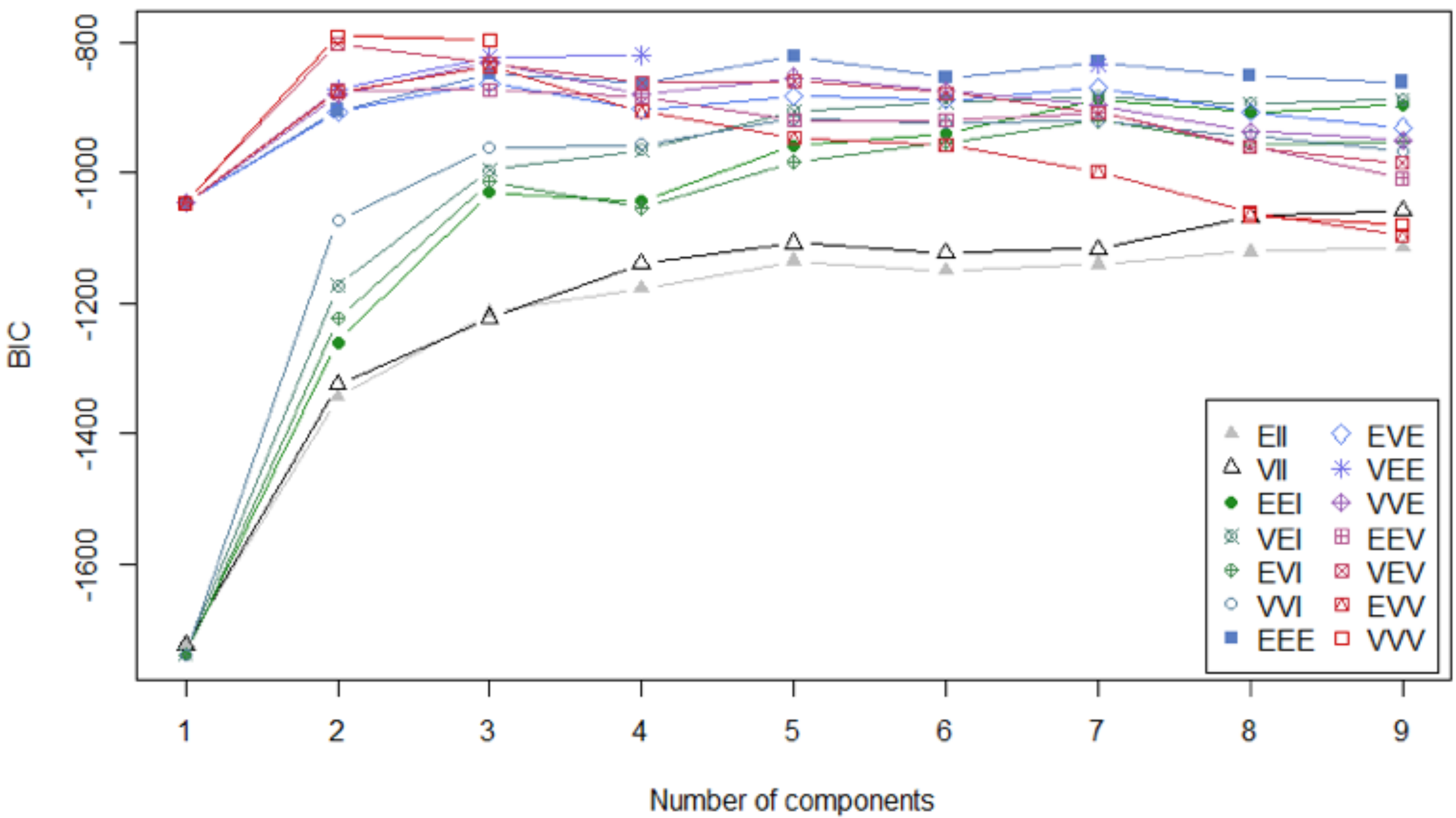
Likelihood of the model

number of parameter of the model

$$BIC(M) = \ln(L(\mathbf{X}, M)) - \frac{1}{2}\#(M)\ln(\underbrace{n}_{\text{number of data points}})$$

number of data points

PM, model evaluation, comparison, AIC y BIC



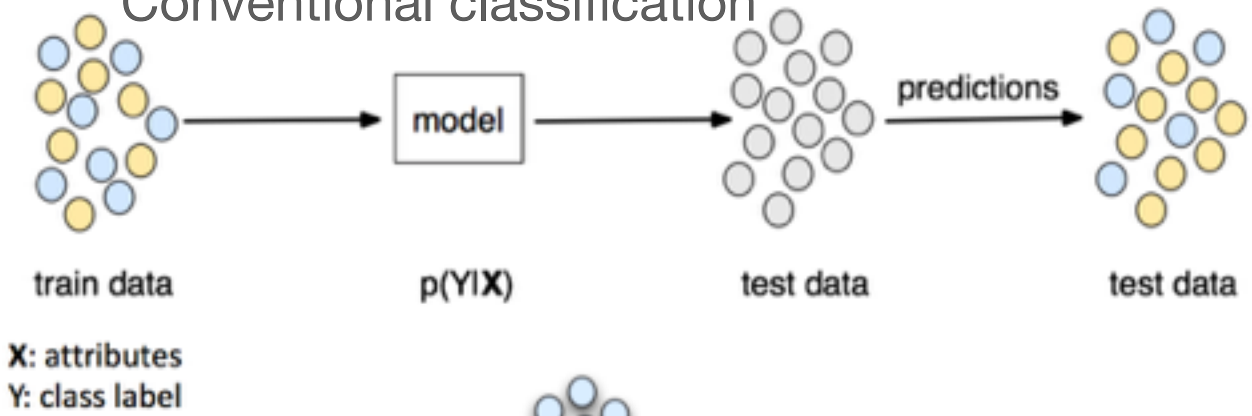
Ensemble methods

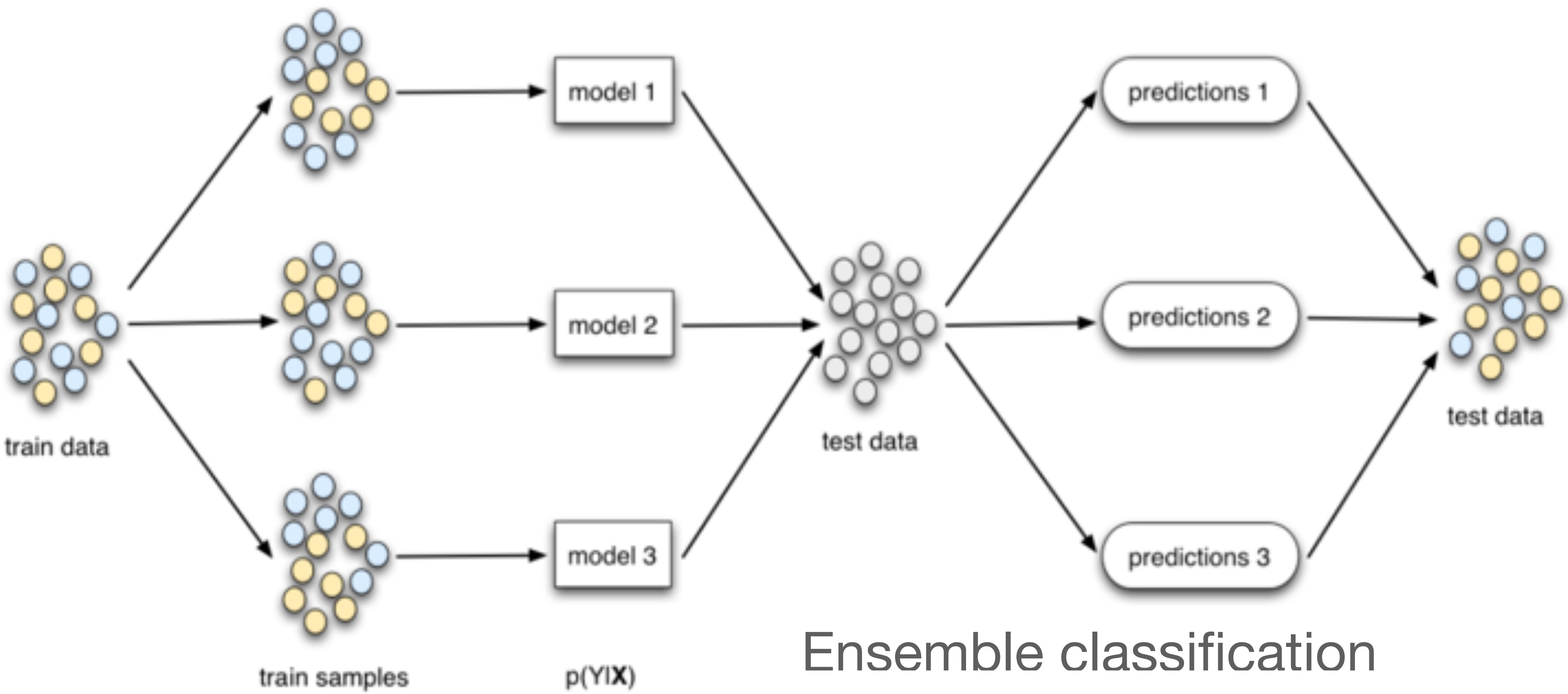
Ensemble methods

- Motivation: Too difficult to construct a single model that optimizes performance
- Approach: Construct many models on different versions of the training set and combine them during prediction
- Goal: reduce bias and/or variance of the error distribution

Ensemble methods

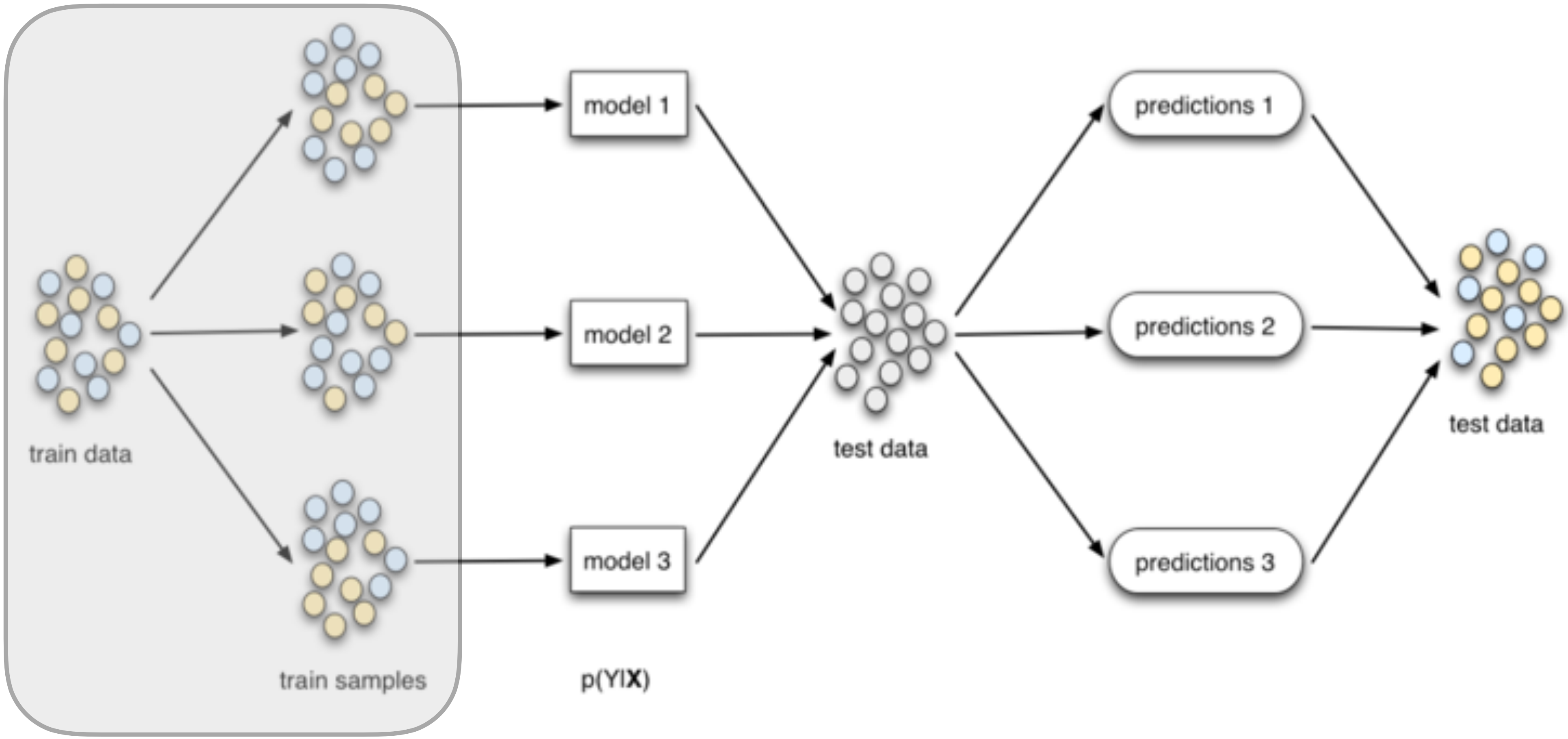
Conventional classification





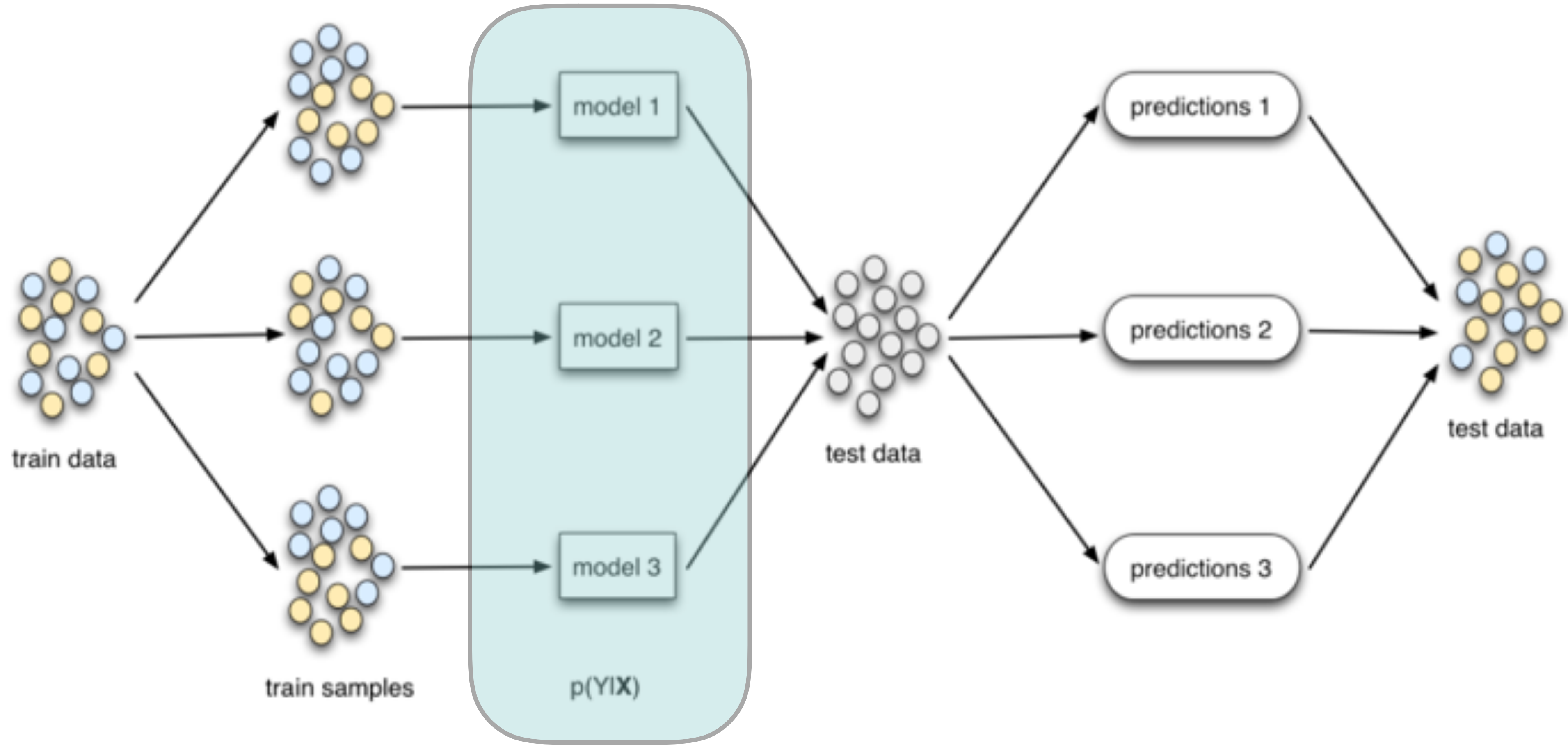
Ensemble classification

Ensemble methods, design



Treatment of input data:
sampling

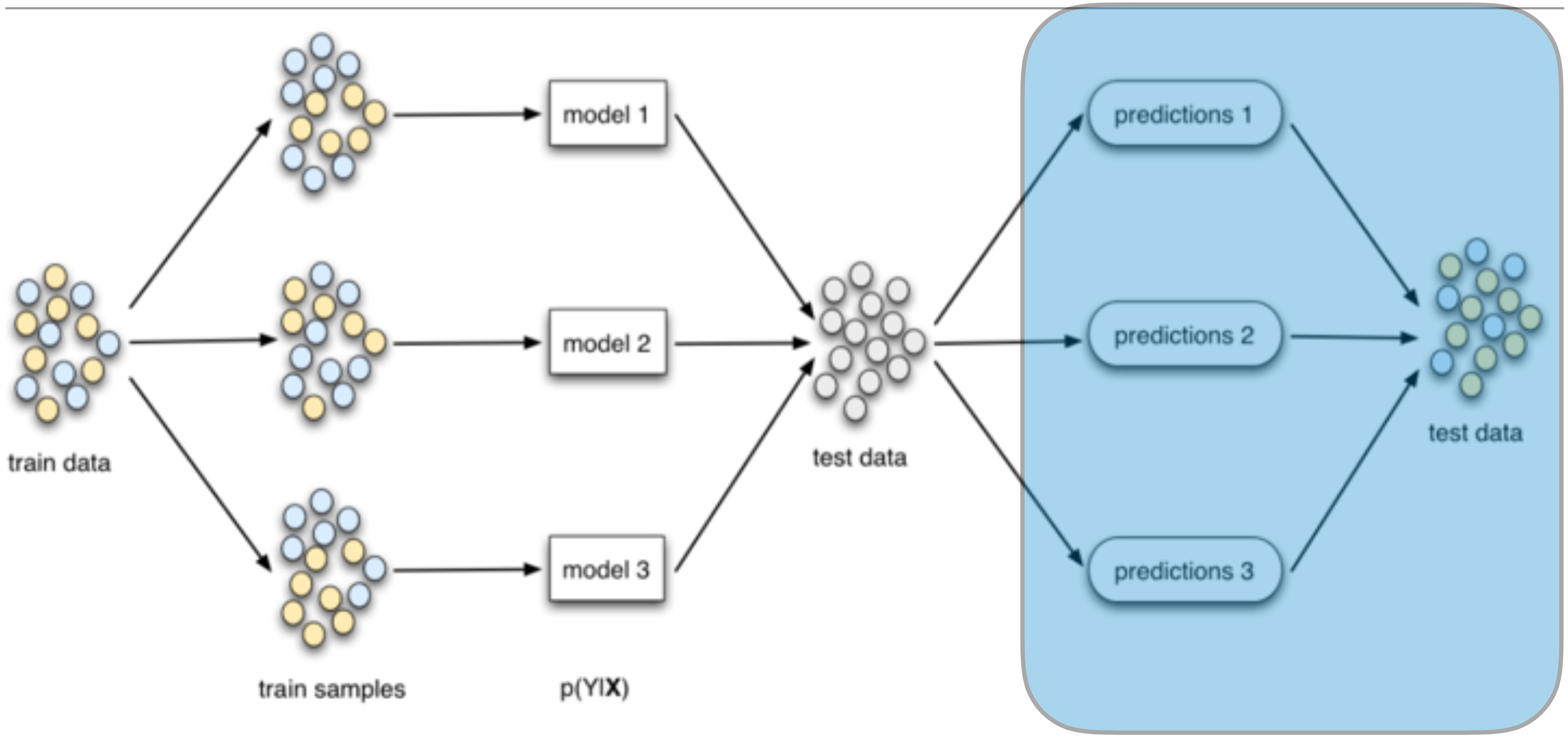
Ensemble methods, design



Treatment of input data:
sampling

Choice of base classifier:
Decision tree
Naive bayes

Ensemble methods, design



Treatment of input data:
sampling

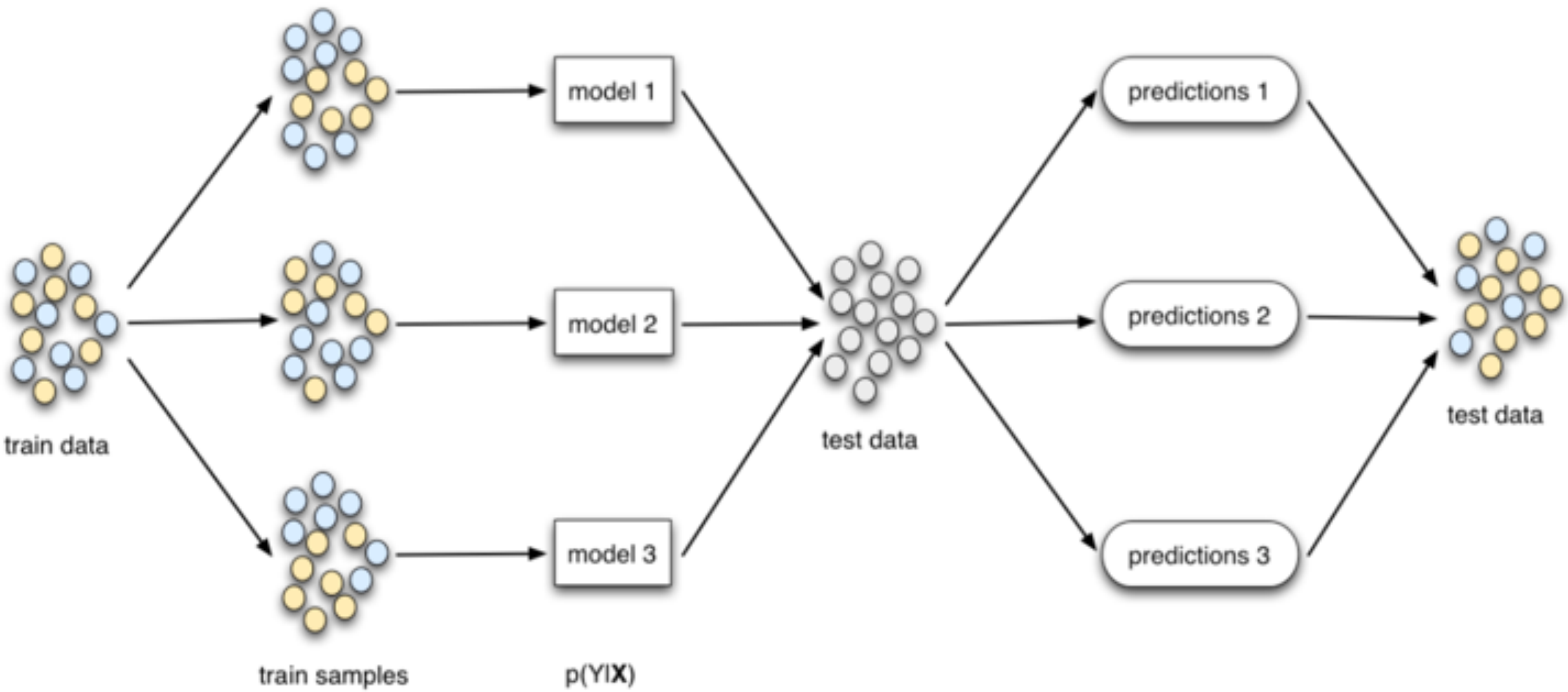
Choice of base classifier:
Decision tree
Naive bayes

Prediction aggregation:
averaging
weighted vote

Ensemble methods, bagging

- **B**ootstrap **agg**regat**ing**
- Main assumption:
  - Combining many unstable predictors in an ensemble produces a stable predictor (i.e., reduces variance)
  - Unstable predictor: small changes in training data produces large changes in the model (e.g., trees)
- Model space: non-parametric, can model any function if an appropriate base model is used

Ensemble methods, bagging



Treatment of input data:
sampling with replacement

Choice of base classifier:
unstable predictor
e.g. Decision tree

Prediction aggregation:
averaging
majority

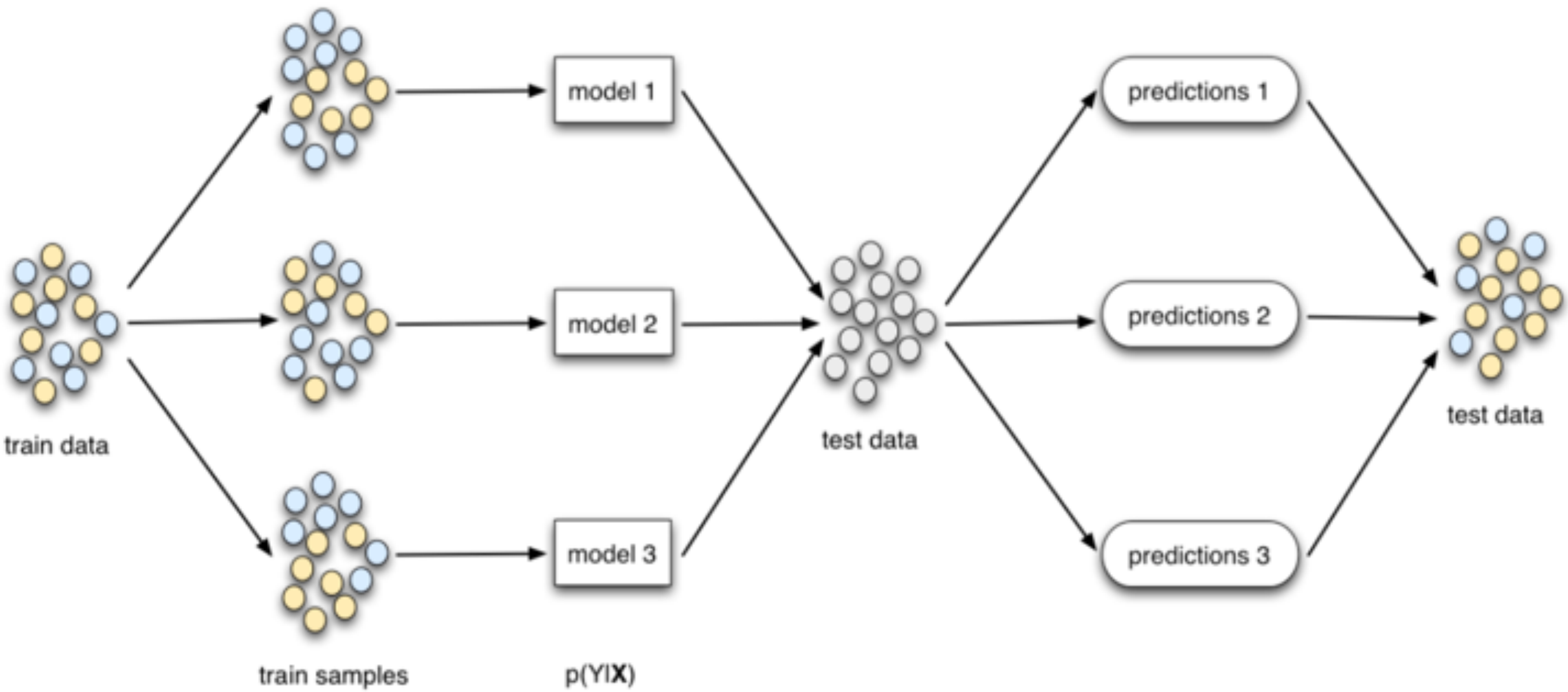
Ensemble methods, bagging

- Given a training data set $D=\{(x_1,y_1),..., (x_N,y_N)\}$, and a M number of models
- For m=1 to M
    - Obtain a bootstrap sample D_m by drawing N instances **with replacement** from D
    - Learn model M_m from D_m
- To classify test instance t, apply each model M_m to t and use majority predication or average prediction
- Models have uncorrelated errors due to difference in training sets (each bootstrap sample has ~68% of D)

Ensemble methods, boosting

- Main assumption:
    - Combining many weak (but stable) predictors in an ensemble produces a strong predictor (i.e., reduces bias)
    - Weak predictor: only weakly predicts correct class of instances (e.g., tree stumps, 1-R)
- Model space: non-parametric, can model any function if an appropriate base model is used

Ensemble methods, boosting



Treatment of input data:
reweight examples

Choice of base classifier:
unstable predictor
e.g. Decision tree

Prediction aggregation:
weighted vote

Ensemble methods, boosting

- Assign every example in training data set $D=\{(x_1,y_1),..., (x_N,y_N)\}$, an equal weight 1/N ($D_1$ corresponds to the original data training set)

- For m=1 to M

    - Learn model M_m from D_m

    - Calculate the error of M_m and up-weight the examples that are incorrectly classified to form D_{m+1}

    - Normalize weights in D_{m+1} to sum to 1

    - Set weight $w_m = \log((1\text{-}error_m)/error_m)$

- To classify test instance t, apply each model M_m to t and take weighted vote of predictions (ie. using w_m)