Machine Learning

Aradhana Behura
BTECH:VSSUT, MTECH:VSSUT, BURLA,
PhD:NIT, RKL

Outline

- Problem solving using Computer
  - Solvability and Tractability
- AI Vs ML Vs DL
- History of AI
- Types of Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Working Principle of Supervised Learning
  - Dataset
  - Splitting dataset into Training, Validation, and Test Sets
  - Classification and Regression
- Performance Evaluation

Can all problems be solvable

- Can all real world problems be translated to mathematical problem: No
  - How to became a topper
- Can all mathematical problems be solved by computer: No
  - Halting problem
- Can all solvable problems be solved in polynomial time or reasonably less time: No
  - Traveling salesman problem

Can all problems be solvable? Continued…

- ❖ What to do if a problem is not solvable ?

- ➢ Settle for solution with some inaccuracy

- ➢ Mimic how nature is solving problems

- ➢ Keep attempting to reduce inaccuracy by evolving the model that mimics nature

What is Machine Learning?

- Machine learning (Ml) is a branch of Artificial Intelligence (AI) that provides computers the ability to automatically learn and improve from experience.

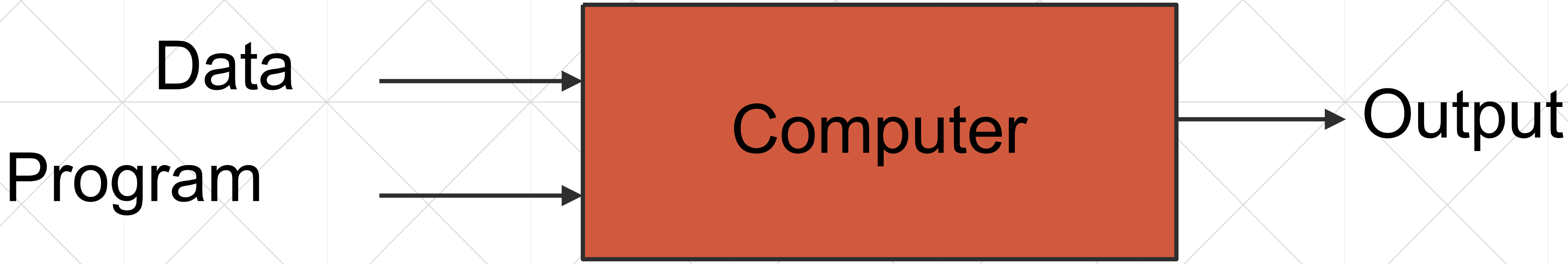
- Machine learning is about designing algorithms that allow a computer to learn from data and experience.

- Machine learning is a data driven model to derives rules from data.

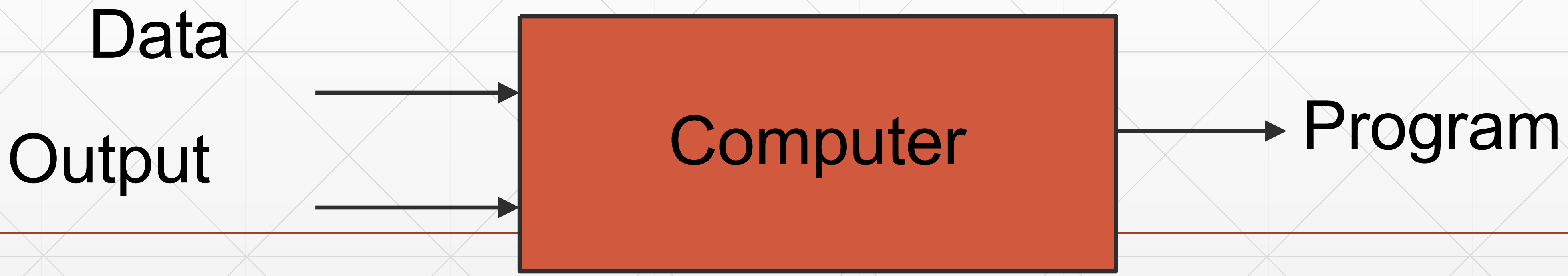
- Machine learning allows computer programs to automatically improve through experience.

Machine Learning Vs Traditional Programming

Traditional Programming



Machine Learning



Difference bet ween AI and Ml

AI	ML
• AI is not a system, but it can be implemented on system to make system intelligent.	• ML is a system that can extract knowledge from dataset.
• AI is used in decision making.	• ML is used in learning from experience.
• AI leads to wisdom or intelligence.	• ML leads to knowledge or experience.
• AI comprises a set of rules which are derived based on parameters estimated using ML.	• ML is one of the way to achieve AI.

Artificial Intelligence (AI) Vs Machine Learning (ML) Vs Deep Learning (DL)

- Objective of AI: Mimic the process of problem solving by natural intelligence

- Objective of ML: Mimic the process of problem solving by human intelligence
  - Learning algorithm with data can improve performance of the machine

- Objective of DL: Extension of one of the ML technique namely Artificial Neural Network (ANN) by significantly increasing more number of hidden layers

Dataset

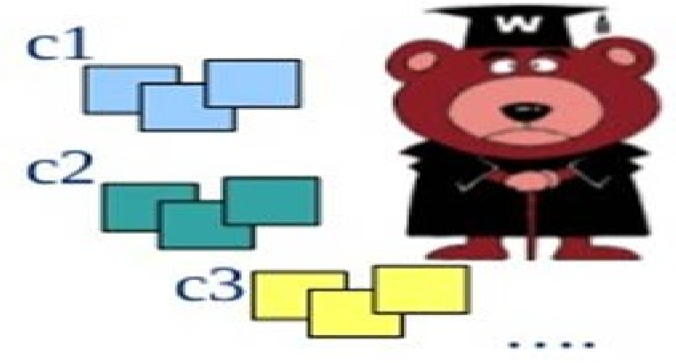
TRS_DT	TRS_TYP_CD	REF_DT	REF_NUM	CO_CD	GDS_CD	QTY	UT_CD	UT_PRIC
21/05/93	00001	04/05/93	25119	10002J	001M	10	CTN	22.000
21/05/93	00001	05/05/93	25124	10002J	032J	200	DOZ	1.370
21/05/93	00001	05/05/93	25124	10002J	033Q	500	DOZ	1.000
21/05/93	00001	13/05/93	25217	10002J	024K	5	CTN	21.000
21/05/93	00001	13/05/93	25216	10026H	006C	20	CTN	69.000
21/05/93	00001	13/05/93	25216	10026H	008Q	10	CTN	114.000
21/05/93	00001	14/05/93	25232	10026H	006C	10	CTN	69.000
21/05/93	00001	14/05/93	25235	10027E	003A	5	CTN	24.000
21/05/93	00001	14/05/93	25235	10027E	001M	5	CTN	24.000
21/05/93	00001	22/04/93	24974	10035E	009F	50	CTN	118.000
21/05/93	00001	27/04/93	25033	10035E	015A	375	GRS	72.000
21/05/93	00001	20/05/93	25313	10041Q	010F	10	CTN	26.000
21/05/93	00001	12/05/93	25197	10054R	002E	25	CTN	24.000

What is Learning?

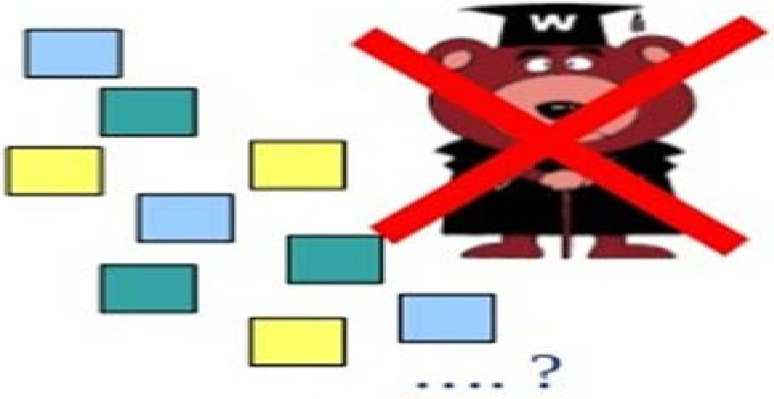
- Learning is a process by which a system improves its performance from experience.
- Arthur Lee Samuel coned the term Machine Learning in 1959. Machine Learning (ML) is a category of an algorithm that enables the computers to learn from data, and even improve themselves, without being explicitly programmed.
- Machine learning can be classified into 3 types of algorithms
  - Supervised Learning (Mimicking child learning from parents/teachers)
  - Unsupervised Learning (Mimicking human learning based on some similar and different objects)
  - Reinforcement Learning (Mimicking human learning based on reward/punishment given by environment)

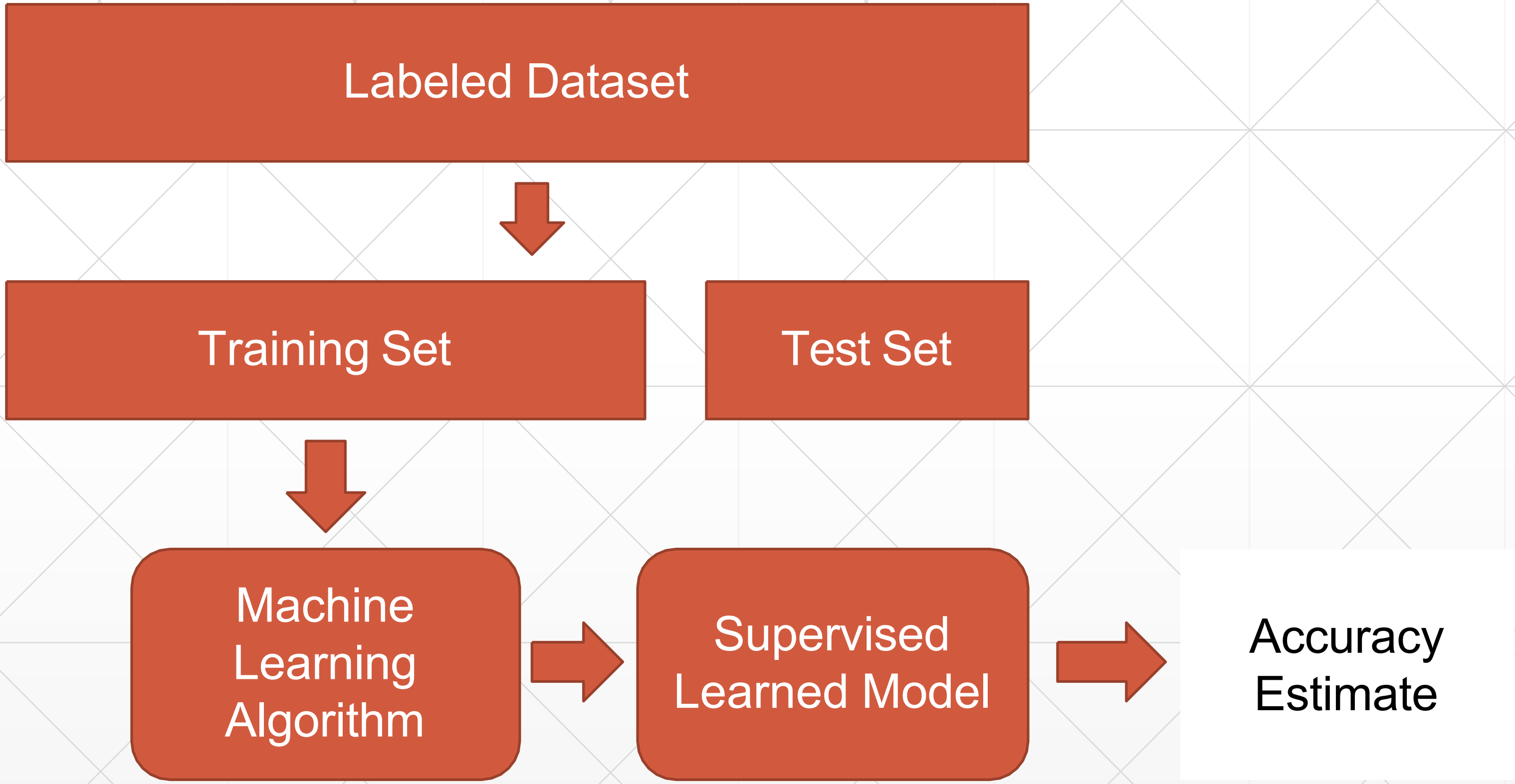
Supervised Vs. Unsupervised

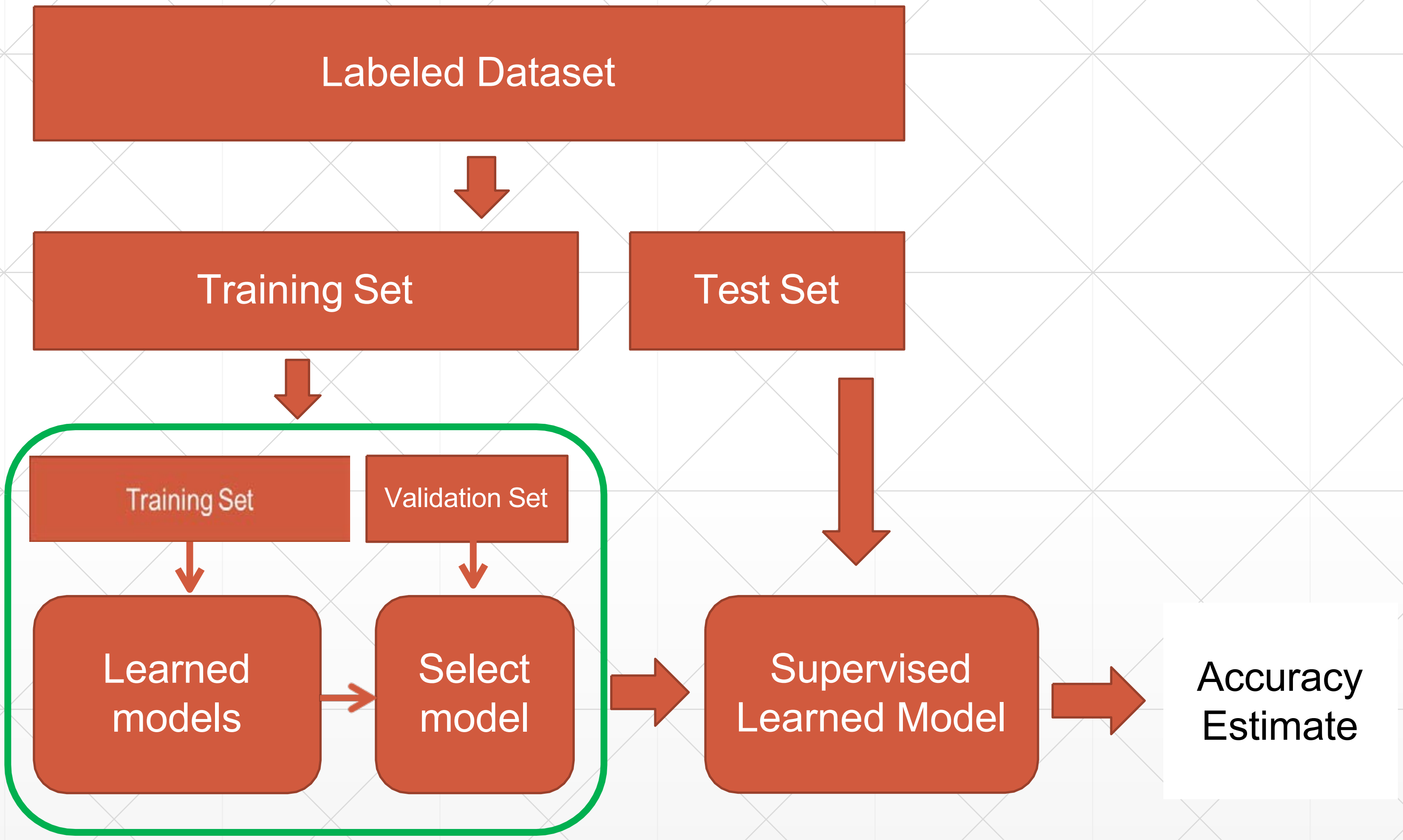
- **Supervised**
  - **knowledge of output -** learning with the presence of an "expert" / teacher
    - data is **labelled** with a class or value
    - **Goal:** predict class or value label
      - e.g. Neural Network, Support Vector Machines, Decision Trees, Bayesian Classifiers ....



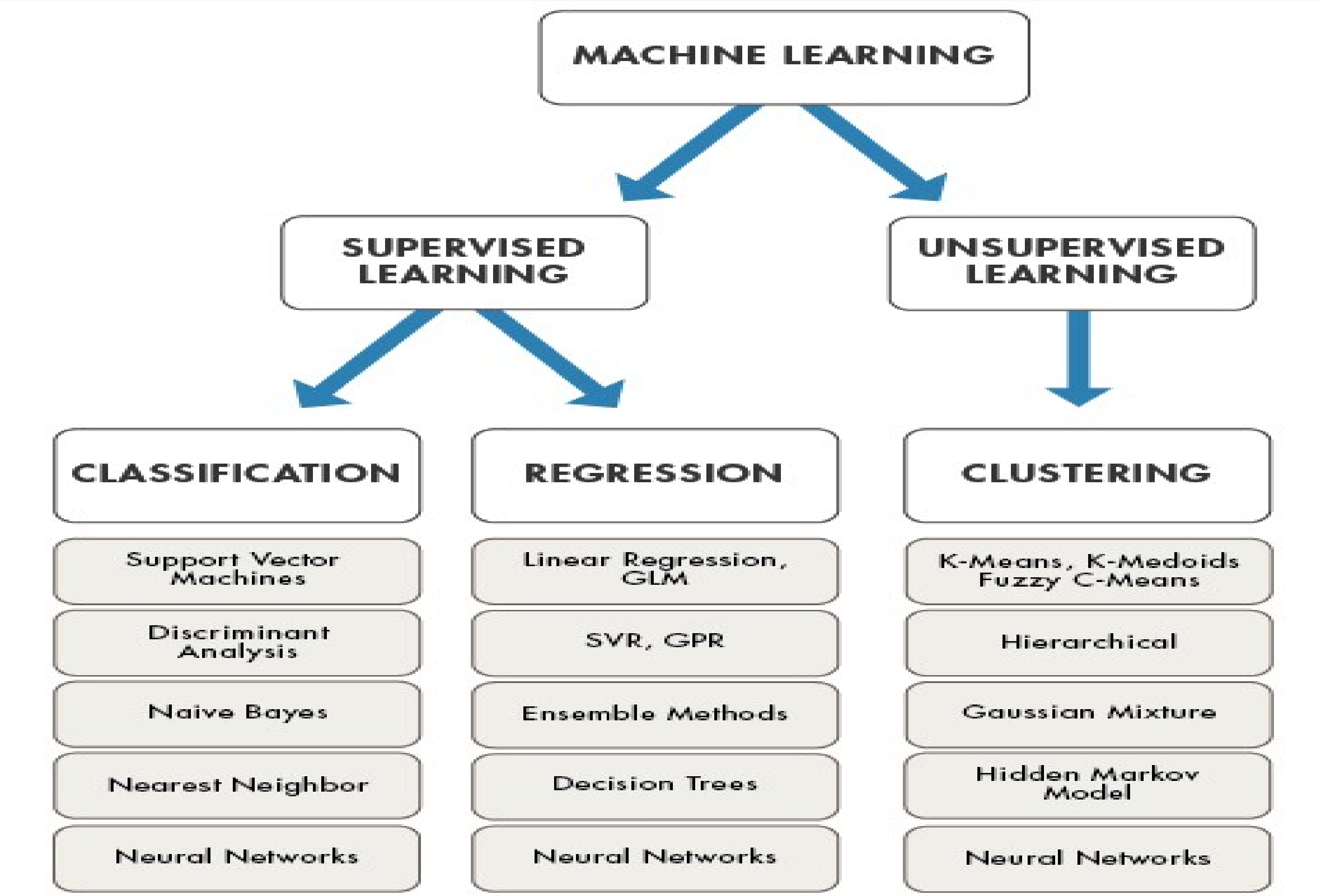
- **Unsupervised**
  - **no knowledge of output** class or value
    - data is **unlabelled** or value un-known
    - **Goal:** determine data patterns/groupings
  - Self-guided learning algorithm
      - (internal self-evaluation against some criteria)
      - e.g. k-means, genetic algorithms, clustering approaches ...



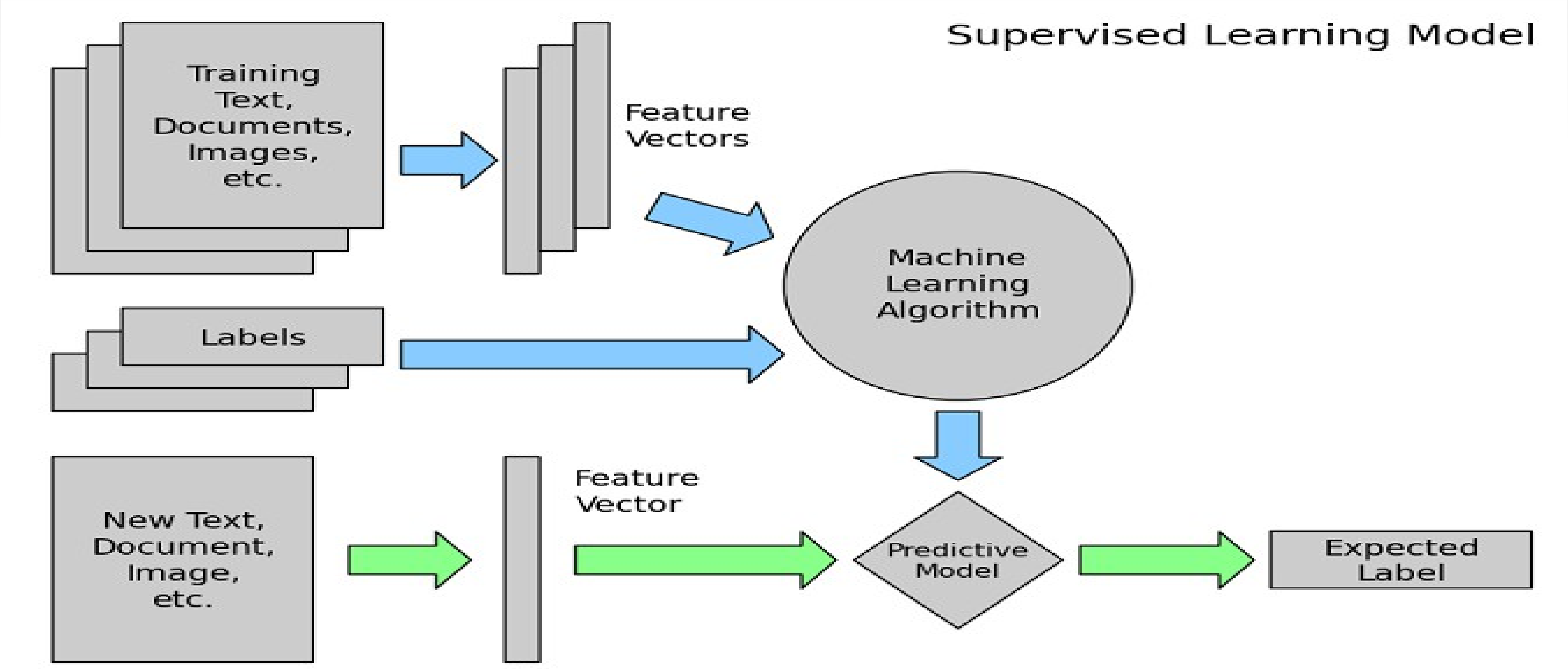

Labeled Dataset
Training Set
Test Set
Machine Learning Algorithm
Supervised Learned Model
Accuracy Estimate


Labeled Dataset
Training Set
Test Set
Training Set
Validation Set
Learned models
Select model
Supervised Learned Model
Accuracy Estimate

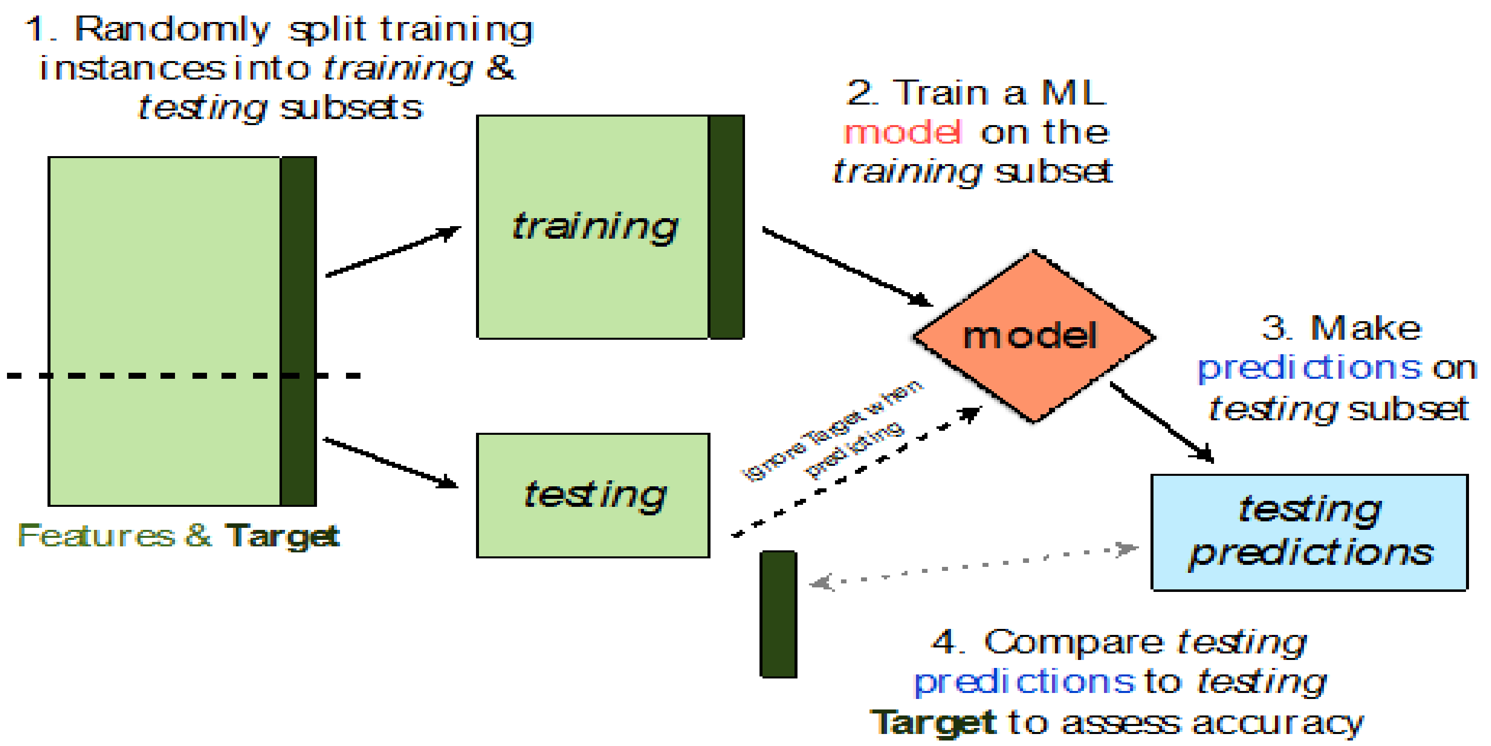
Different Supervised and Unsupervised learning Algorithms



Supervised learning model

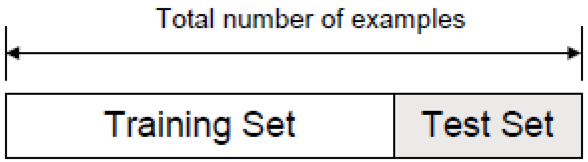


Working principle of supervised learning



Main issues of supervised learning

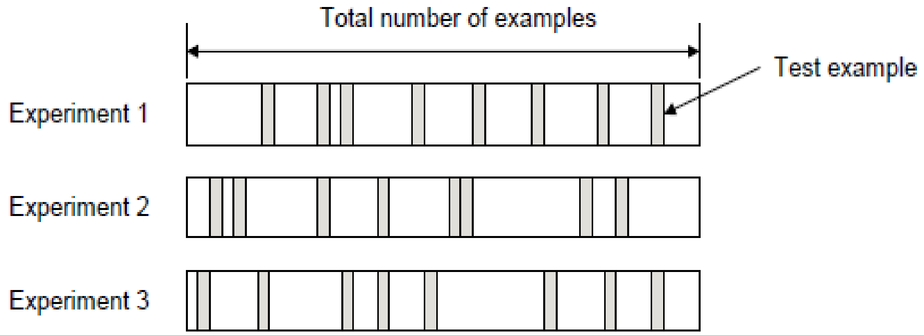
- Splitting dataset into training and test samples
- Selection of classifier
- Two methods are widely used for splitting dataset
  - Holdout method (Naive approach)
  - Cross Validation
- Holdout method
  - Split dataset into two groups
    - **Training set**: used to train the classifier
    - **Test set**: used to estimate the error rate of the trained classifier



- Advantages of holdout method
  - Simple method for splitting dataset (Good for beginners)
  - Computationally take less time for partition of training data and test data
- Disadvantages of holdout method
  - The evaluation may depend heavily on partition of training and test set
  - Increase variance
- The limitations of the holdout can be overcome with a family of resampling methods at the expense of higher computational cost
  - **Cross Validation**
    - Random Subsampling
    - K-Fold Cross-Validation
    - Leave-one-out Cross-Validation

Random Subsampling

- Each data split randomly selects a (fixed) number of examples without replacement
- For each data split we retrain the classifier from scratch with the training examples and then estimate E_i with the test examples

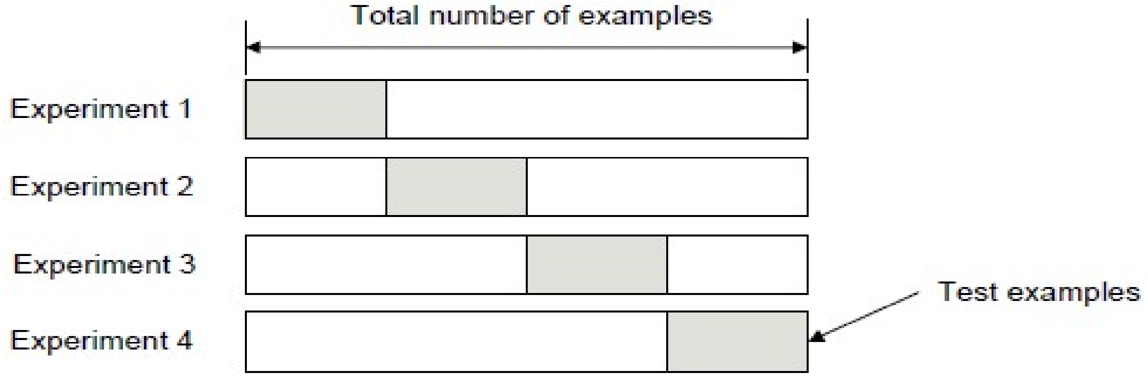


- **The true error estimate is obtained as the average of the separate estimates E_i**
  - This estimate is significantly better than the holdout estimate

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

K-Fold Cross-validation

- **Create a K-fold partition of the the dataset**
  - For each of K experiments, use K-1 folds for training and a different fold for testing
    - This procedure is illustrated in the following figure for K=4

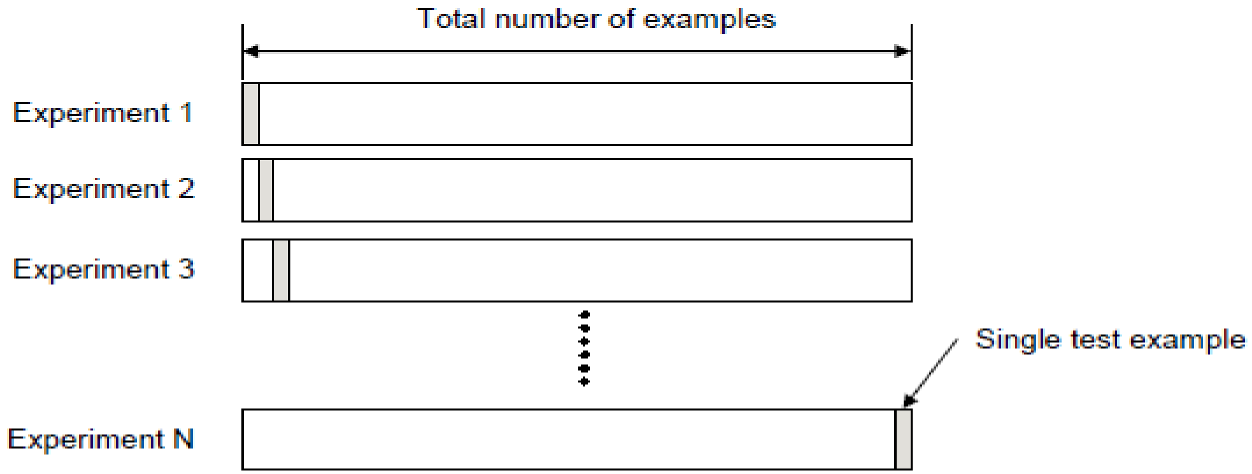


- **K-Fold Cross validation is similar to Random Subsampling**
  - The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing
- **As before, the true error is estimated as the average error rate on test examples**

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

Leave-one-out Cross Validation

- **Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of examples**
  - For a dataset with N examples, perform N experiments
  - For each experiment use N-1 examples for training and the remaining example for testing



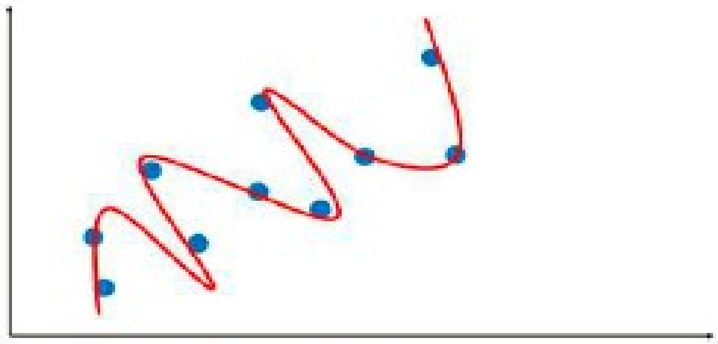
- **As usual, the true error is estimated as the average error rate on test examples**

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i$$

How many folds are needed?

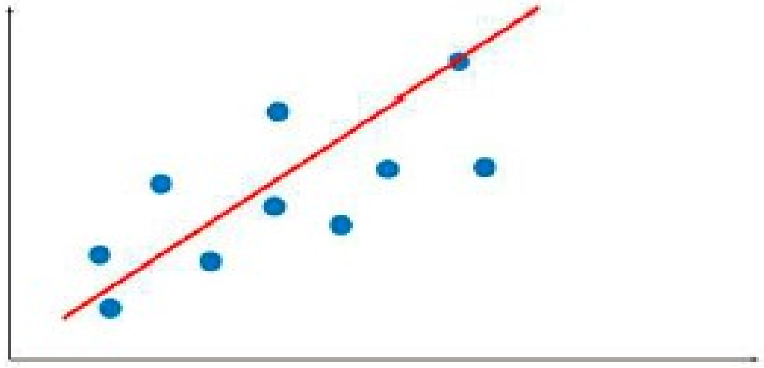
- **With a large number of folds**
  - \+ The bias of the true error rate estimator will be small (the estimator will be very accurate)
  - \- The variance of the true error rate estimator will be large
  - \- The computational time will be very large as well (many experiments)
- **With a small number of folds**
  - \+ The number of experiments and, therefore, computation time are reduced
  - \+ The variance of the estimator will be small
  - \- The bias of the estimator will be large (conservative or smaller than the true error rate)
- **In practice, the choice of the number of folds depends on the size of the dataset**
  - For large datasets, even 3-Fold Cross Validation will be quite accurate
  - For very sparse datasets, we may have to use leave-one-out in order to train on as many examples as possible
- **A common choice for K-Fold Cross Validation is K=10**

Supervised Model Evaluation



Overfitting of Model
(Learned the Training Data but not able to generalize.)

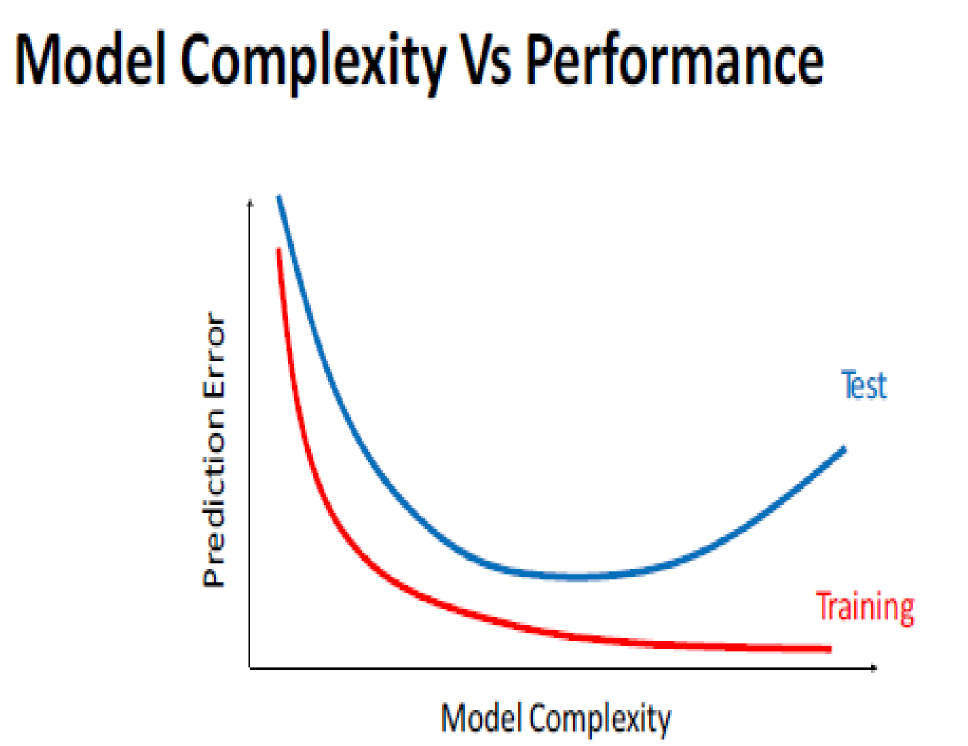
High variance
Low bias

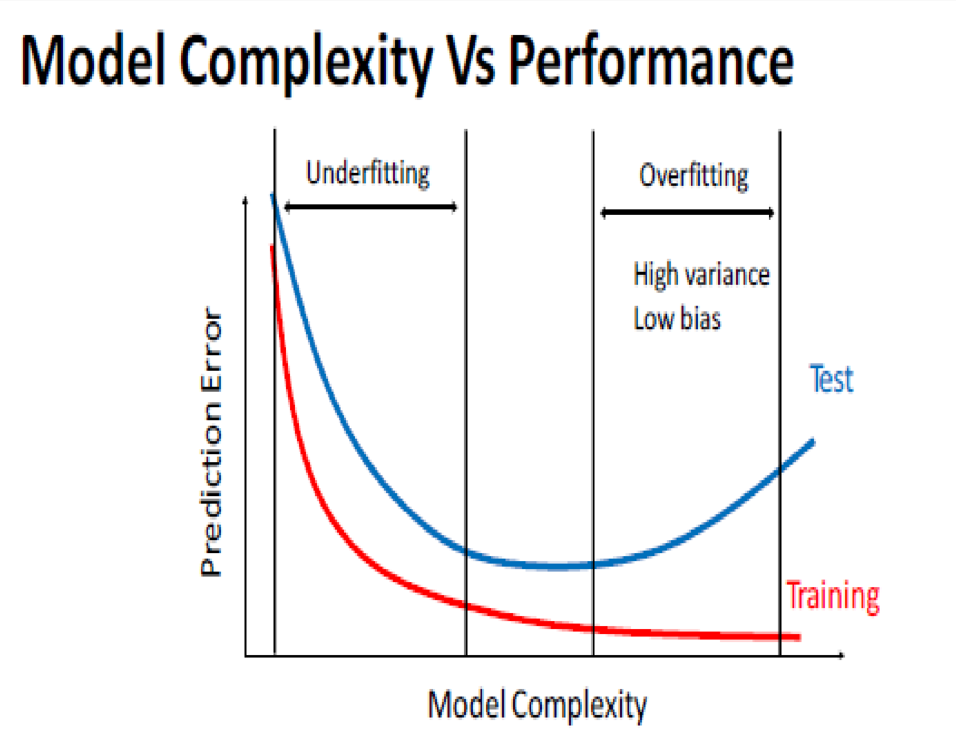


Underfitting of Model
(Model not able to learn the data properly)

Low variance
High bias

Supervised Model Evaluation, Continued…





Model Evaluation and Selection

- Evaluation metrics
  - How can we measure accuracy?
  - Other metrics to consider?
- Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier's accuracy
  - Holdout method
  - Cross-validation
  - Bootstrap (not covered)
- Comparing classifiers:
  - ROC Curves

Classifier Evaluation Metrics: Confusion Matrix

- **Confusion Matrix:**

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

- In a confusion matrix w. *m* classes, $CM_{i,j}$ indicates # of tuples in class *i* that were labeled by the classifier as class *j*

Actual class\Predicted class	play_golf = yes	play_golf = no	Total
play_golf = yes	**6954**	**46**	7000
play_golf = no	**412**	**2588**	3000
Total	7366	2634	10000

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

Real-world truth

Predictions

- **Classifier accuracy,** or recognition rate
  - Percentage of test set tuples that are correctly classified

    Accuracy = (TP + TN)/All
- **Error rate:** *1 – accuracy*, or

    Error rate = (FP + FN)/All

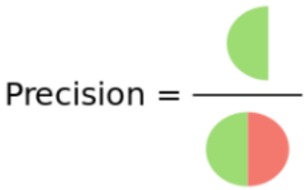
- **Class imbalance problem**
  - One class may be *rare*
    - E.g., fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class
  - Measures handle the class imbalance problem
    - **Sensitivity** (recall): True positive recognition rate
      - **Sensitivity = TP/P**
    - **Specificity**: True negative recognition rate
      - **Specificity = TN/N**

Classifier Evaluation Metrics: Precision and Recall, and F-measures

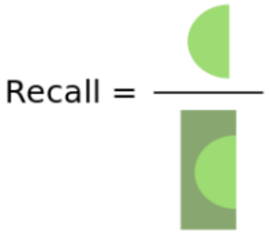
A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P’	**N’**	**All**

- **Precision**: Exactness: what % of tuples that the classifier labeled as positive are actually positive?

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



- **Recall:** Completeness: what % of positive tuples did the classifier label as positive



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

  – Range: [0, 1]

https://en.wikipedia.org/wiki/Precision_and_recall

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- The "inverse" relationship between precision & recall
- ***We want one number to say if a classifier is good or not***
- ***F* measure** (or ***F*-score**): harmonic mean of precision and recall
  - In general, it is the weighted measure of precision & recall

$$F_\beta = \frac{1}{\alpha \cdot \frac{1}{P} + (1 - \alpha) \cdot \frac{1}{R}} = \frac{(\beta^2 + 1) P * R}{\beta^2 P + R}$$

Assigning β times as much weight to recall as to precision

  - ***F1-measure (balanced F-measure)***
    - » That is, when $\beta = 1$,

$$F_1 = \frac{2P * R}{P + R}$$

Classifier Evaluation Metrics: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total
cancer = yes	**90**	**210**	300
cancer = no	**140**	**9560**	9700
Total	230	9770	10000

- Use the same confusion matrix, calculate the precision, recall, F1-score, Sensitivity, Specificity,Accuracy as model evaluation metrics.

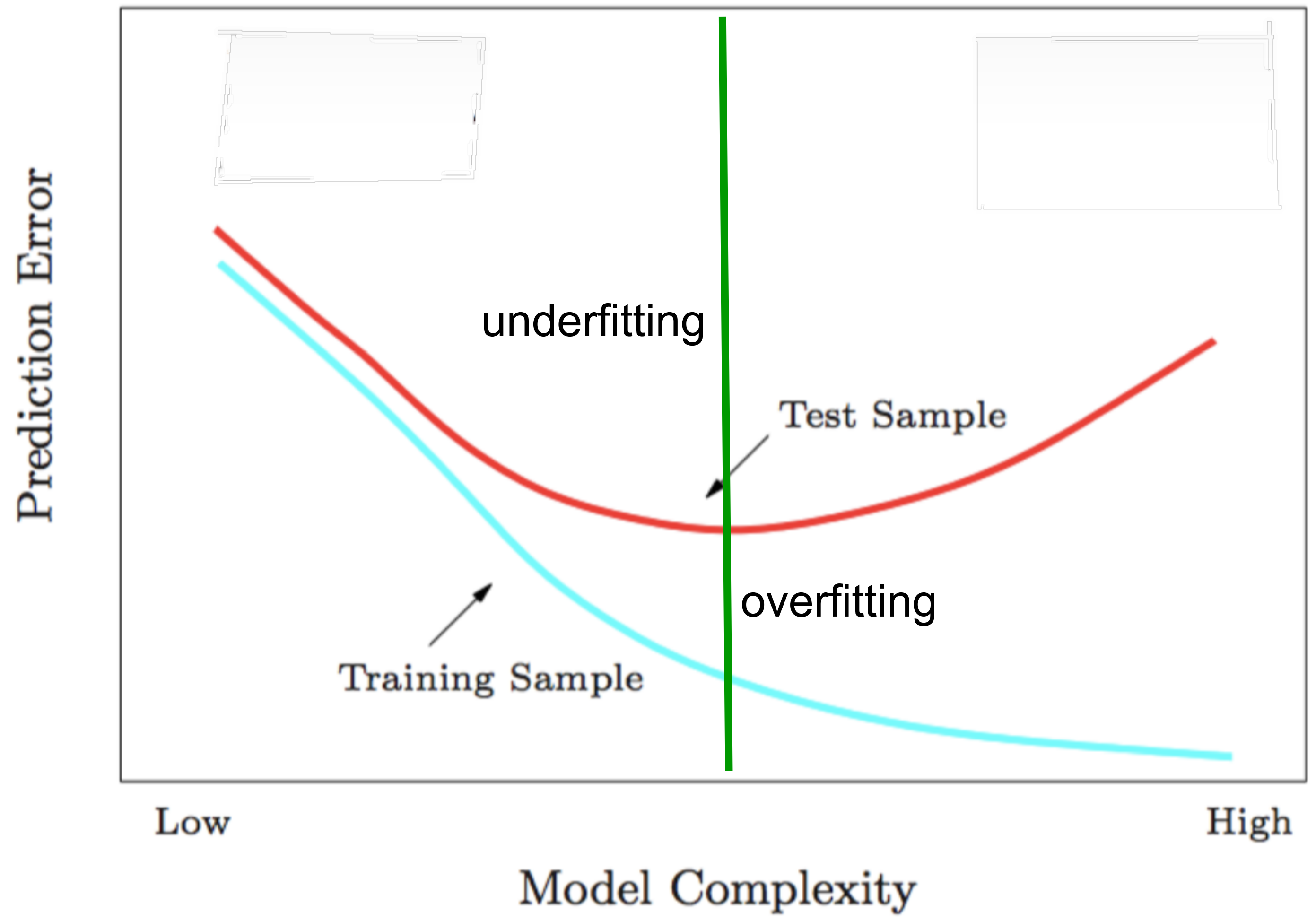
Classifier Evaluation Metrics: Example

- Use the same confusion matrix, calculate the measure just introduced

Actual Class\Predicted class	cancer = yes	cancer = no	Total
cancer = yes	**90**	**210**	300
cancer = no	**140**	**9560**	9700
Total	230	9770	10000

- Sensitivity = TP/P = 90/300 = 30%
- Specificity = TN/N = 9560/9700 = 98.56%
- Accuracy = (TP + TN)/All = (90+9560)/10000 = 96.50%
- Error rate = (FP + FN)/All = (140 + 210)/10000 = 3.50%
- Precision = TP/(TP + FP) = 90/(90 + 140) = 90/230 = 39.13%
- Recall = TP/ (TP + FN) = 90/(90 + 210) = 90/300 = 30.00%
- F1 = 2 P × R /(P + R) = 2 × 39.13% × 30.00%/(39.13% + 30%) = 33.96%

Training Error VS Testing Error



Classifier Evaluation: Holdout

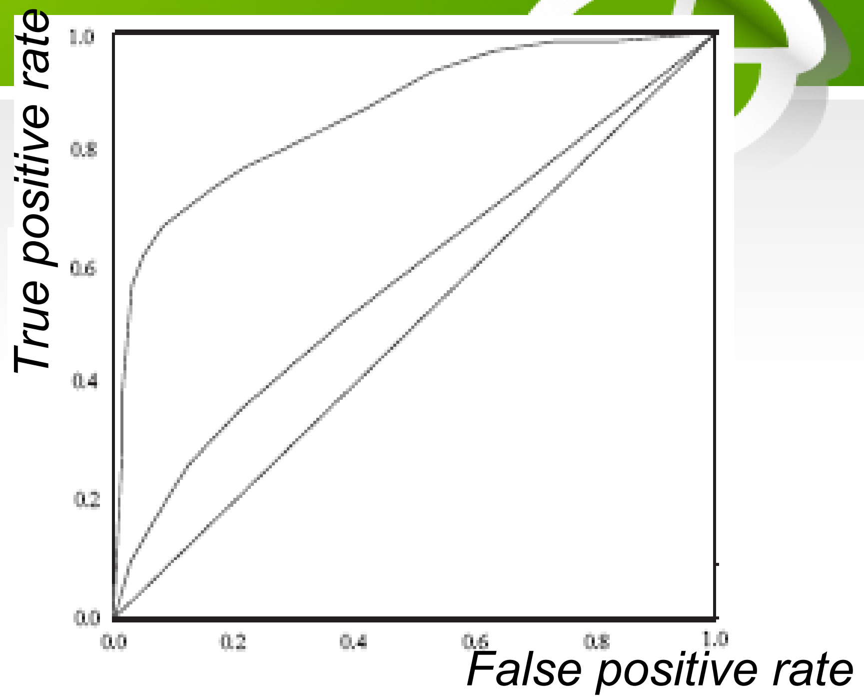
- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Repeated random sub-sampling validation: a variation of holdout
    - Repeat holdout k times, accuracy = avg. of the accuracies obtained

Classifier Evaluation: Cross-Validation

- **Cross-validation** (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - <u>Leave-one-out</u>: *k* folds where *k* = # of tuples, for small sized data
  - <u>***Stratified cross-validation***</u>: folds are stratified so that class distribution, in each fold is approximately the same as that in the initial data

Model Selection: ROC Curves

- **ROC** (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve (**AUC**: Area Under Curve) is a measure of the accuracy of the model
- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



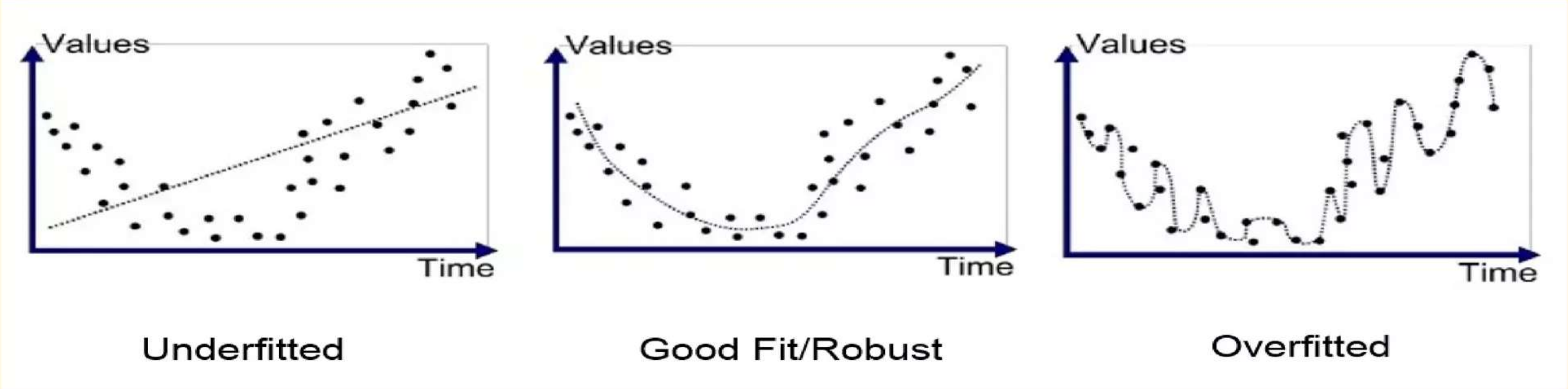
- ❑ Vertical axis represents the true positive rate (TP/P)
- ❑ Horizontal axis rep. the false positive rate (FP/N)
- ❑ The plot also shows a diagonal line
- ❑ A model with perfect accuracy will have an area of 1.0

Bias, Variance and Bias-variance trade-off

Introduction to under fitting and over fitting and best fit

Regression problem: Underfitting, overfitting, best fitting

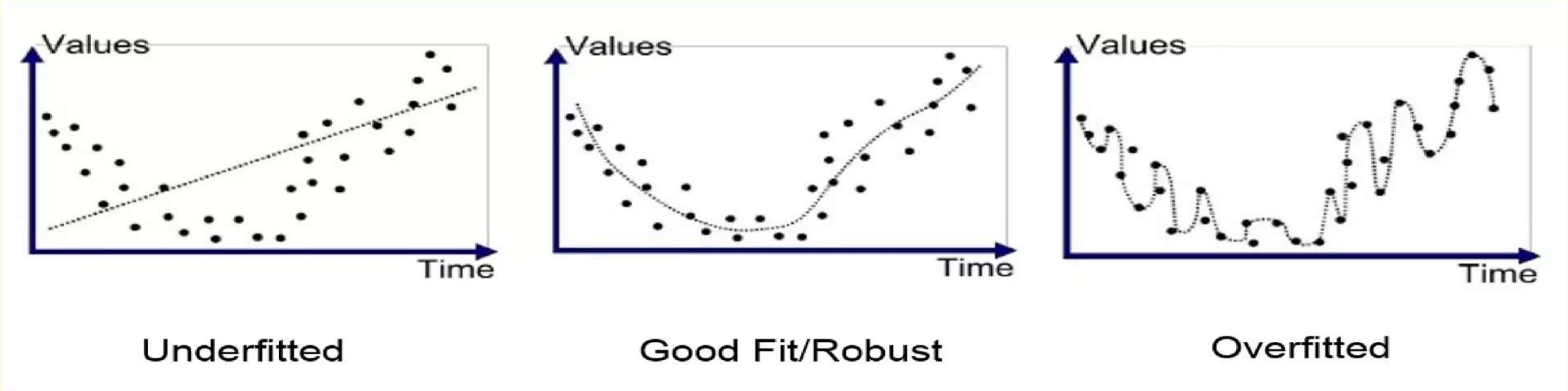
1. **Underfitting:** A model or a ML algorithm is said to have underfitting when it cannot capture the underlying trend of the data.
2. **Overfitting:** A model or a ML algorithm is said to have overfitting when it capture the underlying trend of the data very accurately.
3. **Best fit:** A model or a ML algorithm is said to have best fit when it capture the underlying trend of the data moderately.



Bias and variance

1. **Bias:** It shows the degree of randomness of the training data.
   (a) Based on the training data a suitable model may be created for the regression or classification problem.
   (b) Regression: Linear, logistic, polynomial
   (c) Classification: Decision tree, random forest, naive baye's, KNN, SVM
2. **Variance:** It shows the degree of randomness of the testing data.
   (i) Testing data validates the accuracy of a model, that has been made with the help of training data set.
   (ii) Testing data is nothing but the unlabeled or unknown data.

Underfitting, overfitting, best fit and bias, variance



Note:

⇒ The objective of ML algorithm not only fit for the training data but also fit for the testing data.

⇒ In other words, low bias and low variance is the appropriate solution.

Underfitting	Overfitting	Best fit
High bias	Very low bias	Low bias
High variance	High variance	Low variance

A classifier (Decision tree, random forest, naive baye's, KNN, SVM) works on two types of data

- Training data
- Testing data

Example:
Classifier comes under the category of underfitting, overfitting, and best fit based on its training and testing accuracy.

Underfitting	**Overfitting**	**Best fit**
Train error=25%	Train error=1%	Train error=8%
Test error= 27%	Test error= 23%	Test error=9%

Mathematical intuition

$$bias(\hat{f}(x)) = \mathbb{E}[\hat{f}(x)] - f(x) \quad (1)$$

$$variance(\hat{f}(x)) = \mathbb{E}\left[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2\right] \quad (2)$$

⇒ $\hat{f}(x) \rightarrow$ output observed through the training model

⇒ For linear model $\hat{f}(x) = w_1 x + w_0$

⇒ For complex model $\hat{f}(x) = \sum_{i=1}^{p} w_i x^i + w_0$

⇒ We don't have idea regarding the true $f(x)$.

⇒ **Simple model:** Low bias & high variance

⇒ **Complex model:** High bias & low variance

$$\mathbb{E}[(y - \hat{f}(x))^2] = bias^2 + Variance + \sigma^2 \ (Irreducible\ error) \quad (3)$$

Continued–

⇒ Let there be $n+m$ sample data in a given data set in which n and m samples are taken for the training and testing (validation) purpose then

$$train_{err} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

$$test_{err} = \frac{1}{m}\sum_{i=n+1}^{n+m}(y_i - \hat{f}(x_i))^2$$

⇒ If model complexity is increased the training model becomes too optimistic gives a wrong picture of how close $\hat{f}$ to f.

⇒ Let $\mathcal{D} = \{x_i, y_i\}_1^{n+m}$ is a given data set, where

$$y_i = f(x_i) + \epsilon_i$$

⇒ $\epsilon \sim \mathcal{N}(0, \sigma^2)$

Continued–

⇒ We use $\hat{f}$ to approximate f, where training set $T \subset \mathcal{D}$ such that

$$y_i = \hat{f}(x_i)$$

⇒ We are interested to know

$$\mathbb{E}\left[\left(\hat{f}(x_i) - f(x_i)\right)^2\right]$$

⇒ We cannot estimate the above expression directly, because we don't know $f(x_i)$

$$\begin{aligned}
\mathbb{E}\left[(\hat{y}_i - y_i)^2\right] &= \mathbb{E}\left[(\hat{f}(x_i) - f(x_i) - \epsilon_i)^2\right] \Leftarrow y_i = f(x_i) + \epsilon_i \\
&= \mathbb{E}\left[(\hat{f}(x_i) - f(x_i))^2 - 2\epsilon_i(\hat{f}(x_i) - f(x_i)) + \epsilon_i^2\right] \\
&= \mathbb{E}\left[(\hat{f}(x_i) - f(x_i))^2\right] - 2\mathbb{E}\left[\epsilon_i(\hat{f}(x_i) - f(x_i))\right] + \mathbb{E}\left[\epsilon_i^2\right]
\end{aligned}$$

$$\mathbb{E}\left[(\hat{f}(x_i) - f(x_i))^2\right] = \mathbb{E}\left[(\hat{y}_i - y_i)^2\right] + 2\mathbb{E}\left[\epsilon_i(\hat{f}(x_i) - f(x_i))\right] - \mathbb{E}\left[\epsilon_i^2\right] \quad (4)$$

Continued–

⇒ Empirical estimate: Let $Z = \{z_i\}_{i=1}^{n}$

$$\mathbb{E}(Z) = \frac{1}{n}\sum_{i=1}^{n} z_i$$

⇒ We can empirically evaluate R.H.S using training or test observation

Case 1: **Using test observation:**

$$\underbrace{\mathbb{E}\left[(\hat{f}(x_i) - f(x_i))^2\right]}_{True\ error} = \mathbb{E}\left[(\hat{y}_i - y_i)^2\right] + 2\mathbb{E}\left[\epsilon_i(\hat{f}(x_i) - f(x_i))\right] - \mathbb{E}\left[\epsilon_i^2\right]$$

$$= \underbrace{\frac{1}{m}\sum_{i=n+1}^{n+m}(\hat{y}_i - y_i)^2}_{Empirical\ error} - \underbrace{\frac{1}{m}\sum_{i=n+1}^{n+m}(\epsilon_i)^2}_{Small\ constant} + 2\underbrace{\mathbb{E}\left[\epsilon_i(\hat{f}(x_i) - f(x_i))\right]}_{Covariance}$$

$\because Cov(X, Y) = \mathbb{E}\left[(X - \mu_x)(Y - \mu_y)\right]$ let $\epsilon_i = X$ and $Y = (\hat{f}(x_i) - f(x_i))$

Continued–

$\Rightarrow \mathbb{E}\Big[(X-\mu_x)(Y-\mu_y)\Big] = \mathbb{E}\Big[X(Y-\mu_y)\Big] = \mathbb{E}\big[XY\big] - \mu_y\mathbb{E}\big[X\big]$
$= \mathbb{E}[XY]$

$\Rightarrow$ None of the test data participated for estimating the $\hat{f}(x_i)$.

$\Rightarrow$ $\hat{f}(x_i)$ is estimated only using the training data.

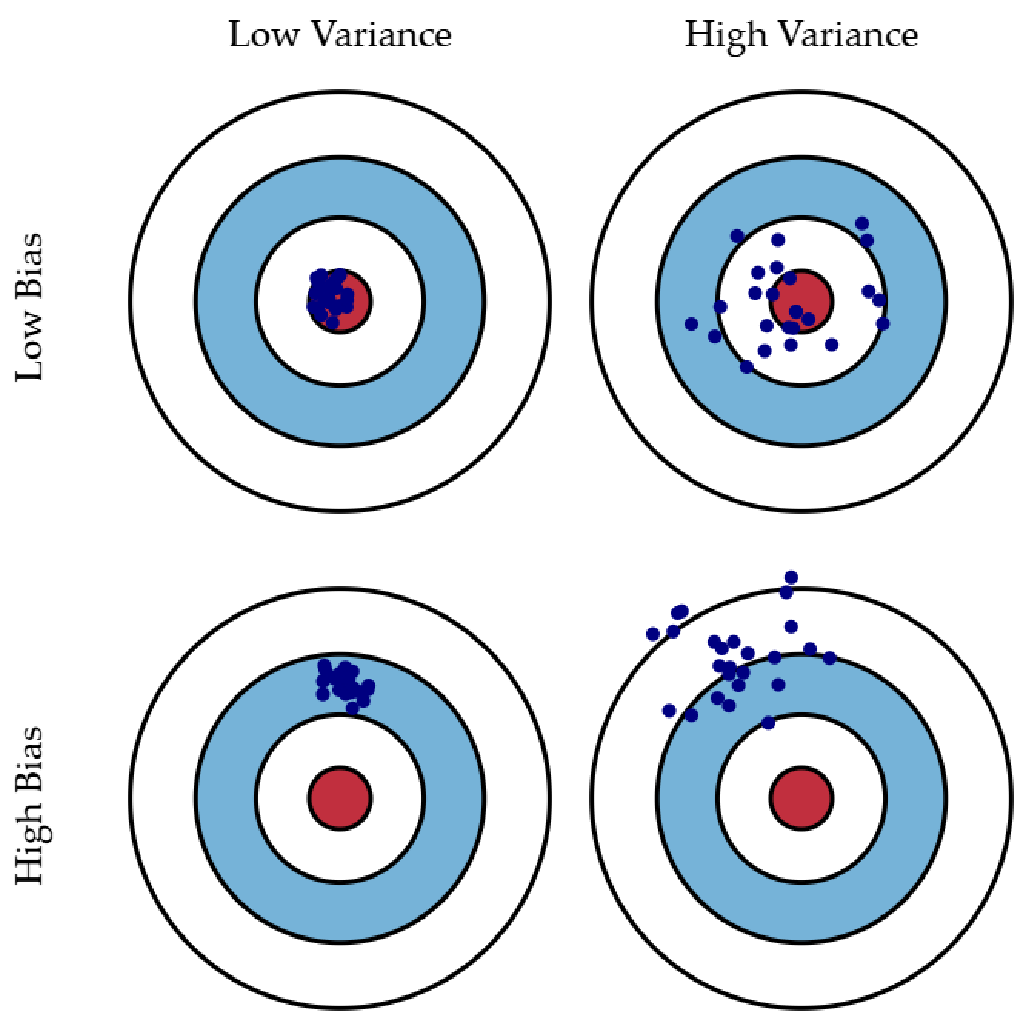
$\Rightarrow \therefore \epsilon_i \perp (\hat{f}(x_i) - f(x_i))$
$\therefore \mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] = 0$

$\Rightarrow$ True error=Empirical error+Small constant $\leftarrow$ Test data

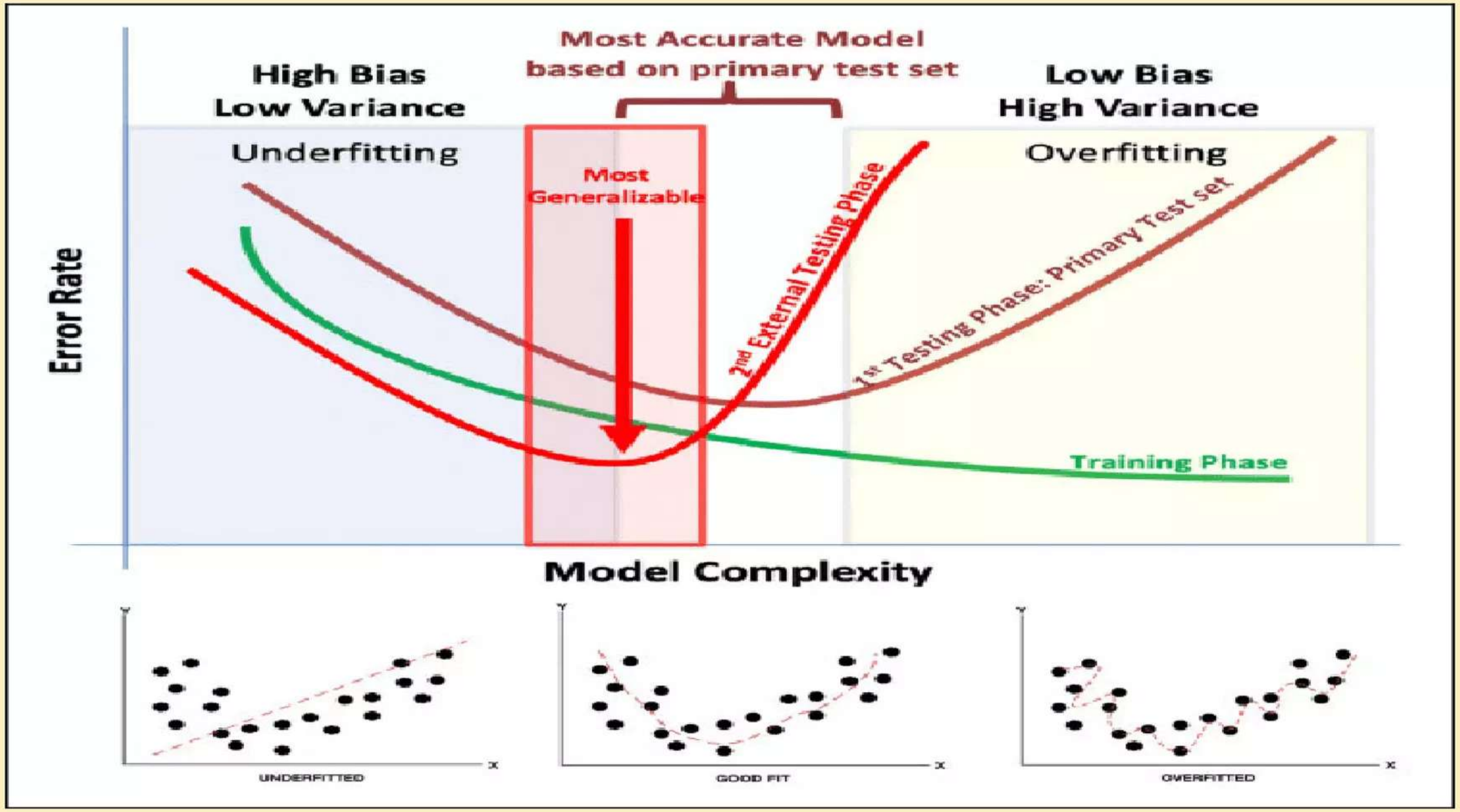
Case 2: **For training observation:**
$\mathbb{E}[XY] \neq 0$
Using Stein's Lemma

$$\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(\hat{f}(x_i) - f(x_i)) = \frac{\sigma^2}{n}\sum_{i=1}^{n}\frac{\partial \hat{f}(x_i)}{\partial y_i}$$

Graphical visualization of bias and variance



Bias and variance trade-off relation



References

E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

J. Grus, *Data science from scratch: first principles with python*. O'Reilly Media, 2019.

T. M. Mitchell, *The discipline of machine learning*. Carnegie Mellon University, School of Computer Science, Machine Learning , 2006, vol. 9.