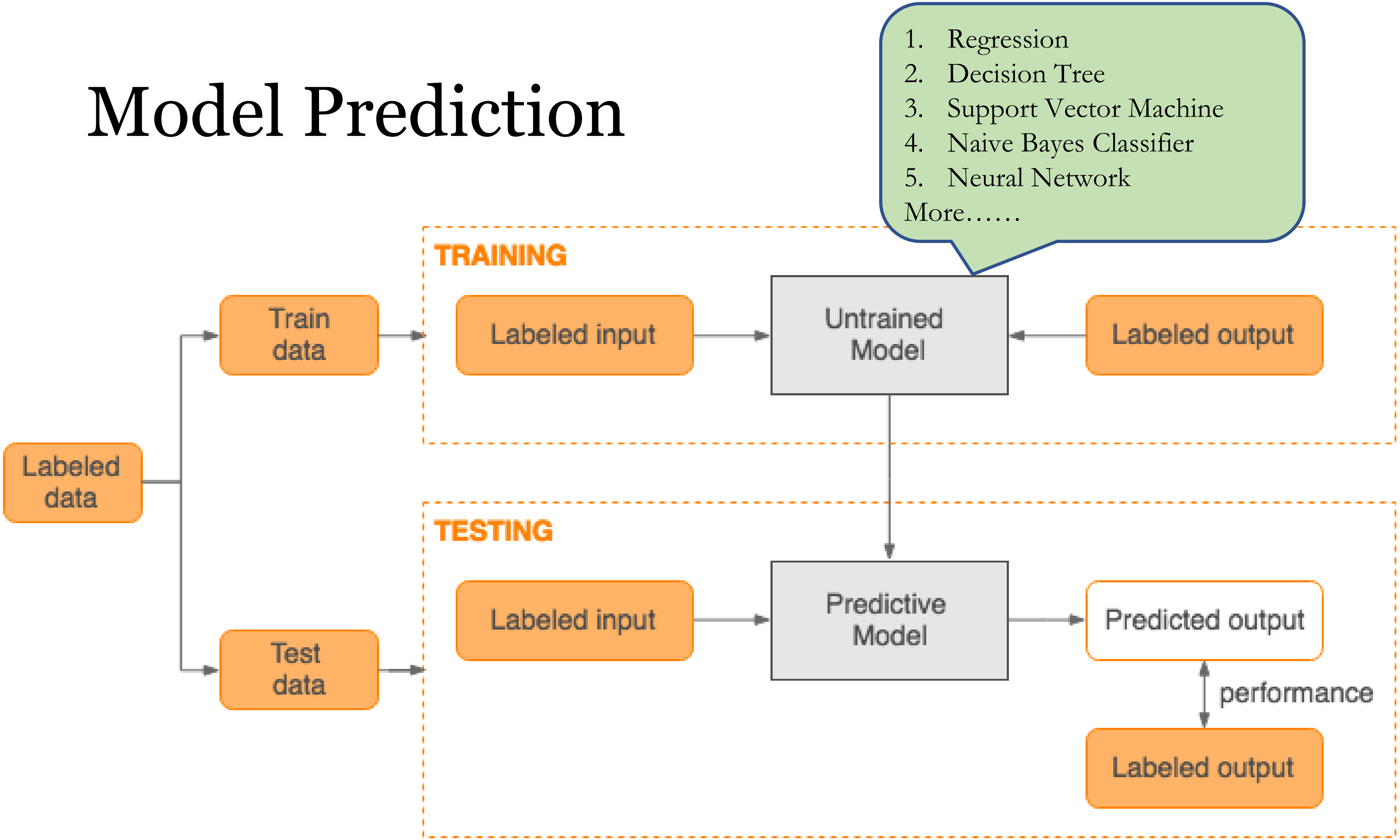
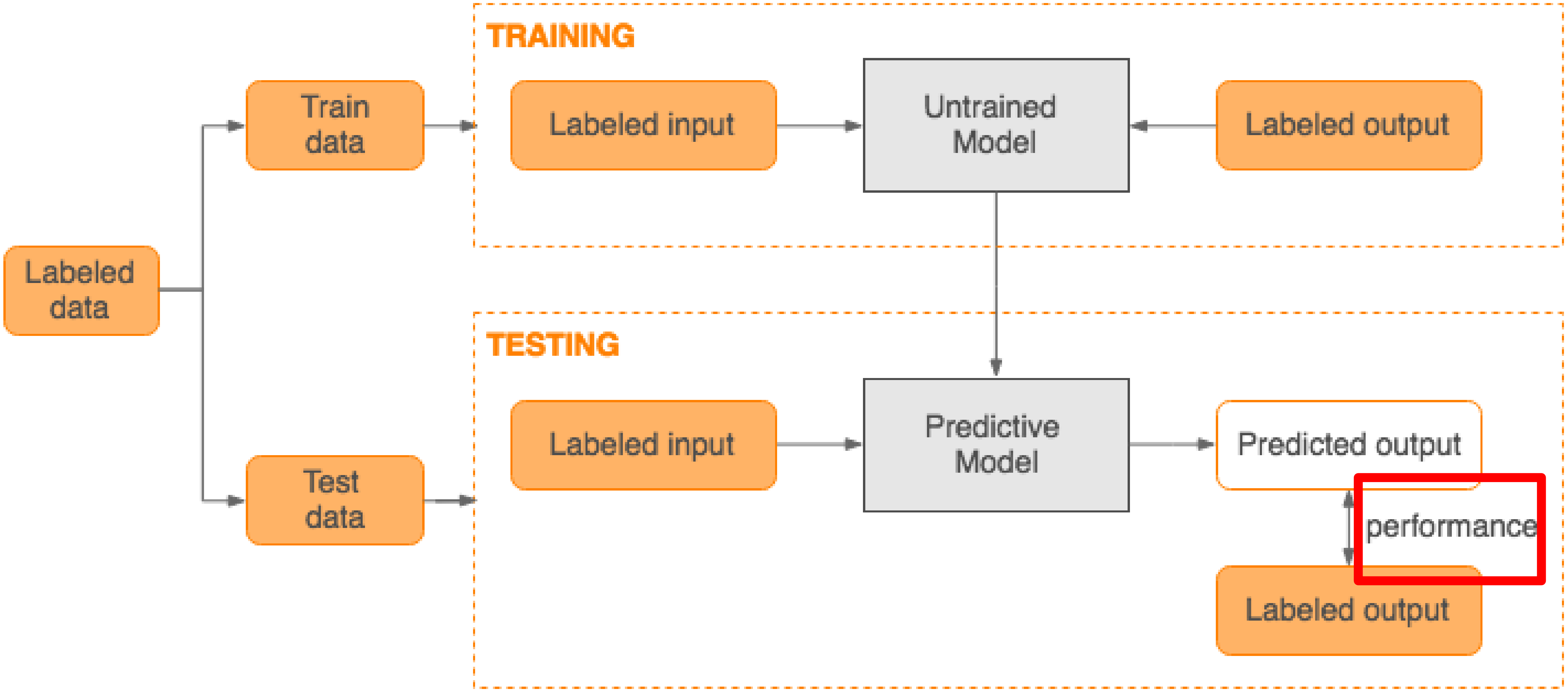
BT2101 Week 6
Some Tips

Model Prediction



Model Prediction



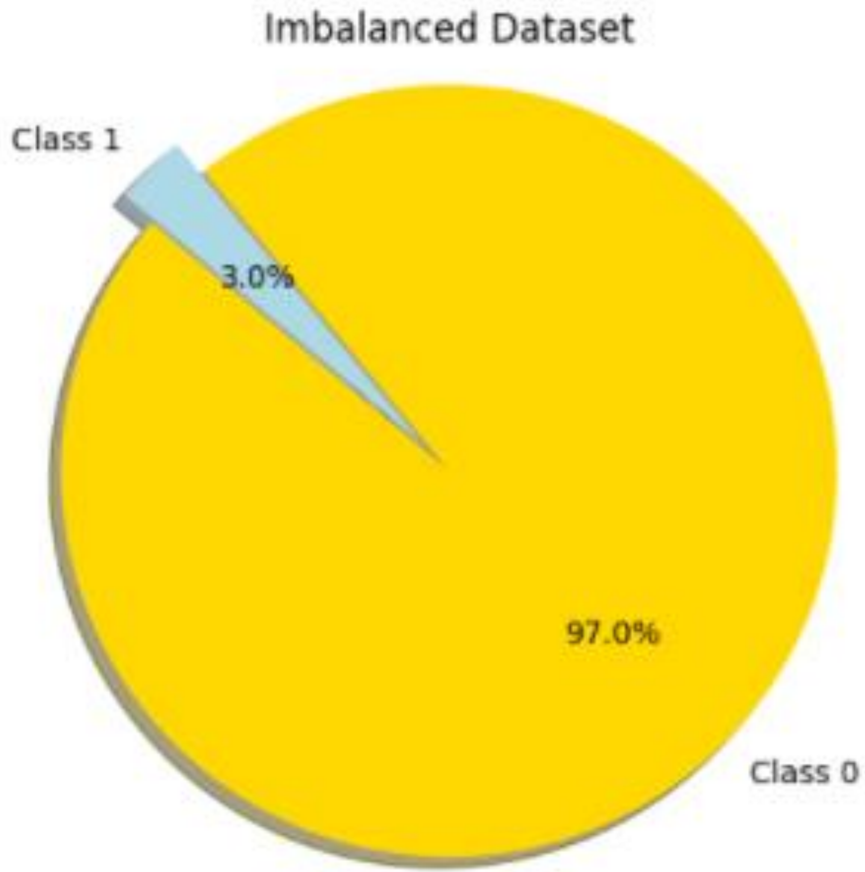
Performance of Binary Classifier

- Problems of **Accuracy/Error** metrics:

- Example: Cancer Detection
  - Class 1 (Positive): Having cancer
  - Class 0 (Negative): Being Healthy

- What if your model misclassified all the "Class 1" cases but correctly classified all the "Class 0" cases?

♦ **Your model accuracy is 97%, but do you think this is a good model?**



Performance of Binary Classifier

- Confusion Matrix:

	Predicted: NO	Predicted: YES
Actual: NO	TN = ??	FP = ??
Actual: YES	FN = ??	TP = ??

Sensitivity (or Recall, TPR) = TP ÷ (TP+FN)

Precision (or PPR, PPV) = TP ÷ (TP+FP)

True Positive Rate or Hit Rate or Recall or Sensitivity or TP Rate	TP/P	The proportion of positive instances that are correctly classified as positive
False Positive Rate or False Alarm Rate or FP Rate	FP/N	The proportion of negative instances that are erroneously classified as positive
False Negative Rate or FN Rate	FN/P	The proportion of positive instances that are erroneously classified as negative $= 1 -$ True Positive Rate

True Negative Rate or Specificity or TN Rate	TN/N	The proportion of negative instances that are correctly classified as negative
Precision or Positive Predictive Value	TP/(TP+FP)	Proportion of instances classified as positive that are really positive
F1 Score	$(2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$	A measure that combines Precision and Recall
Accuracy or Predictive Accuracy	$(TP + TN)/(P + N)$	The proportion of instances that are correctly classified
Error Rate	$(FP + FN)/(P + N)$	The proportion of instances that are incorrectly classified

Sensitivity v.s. Precision

Sensitivity (or Recall, TPR) = TP ÷ (TP+$\boxed{\text{FN}}$)

Precision (or PPR, PPV) = TP ÷ (TP+$\boxed{\text{FP}}$)

Do you care more about **FN** or **FP** ?

Case I: New HIV Test Method

	Predicted: NO	Predicted: YES
Actual: NO	TN=900	FP=0
Actual: YES	FN=90	TP=10

- 1000 people: 100 are real HIV patients, 900 are healthy people
- Accuracy = (900+10)/1000=91%
- Sensitivity = 10/(10+90) = 10% ♣
- Precision = 1/(1+0) = 100%

♥ **FN in HIV test kills people !**

Case II: Advertising Target on Credit Card Users

	Predicted: NO	Predicted: YES
Actual: NO	TN=909	FP=81
Actual: YES	FN=1	TP=9

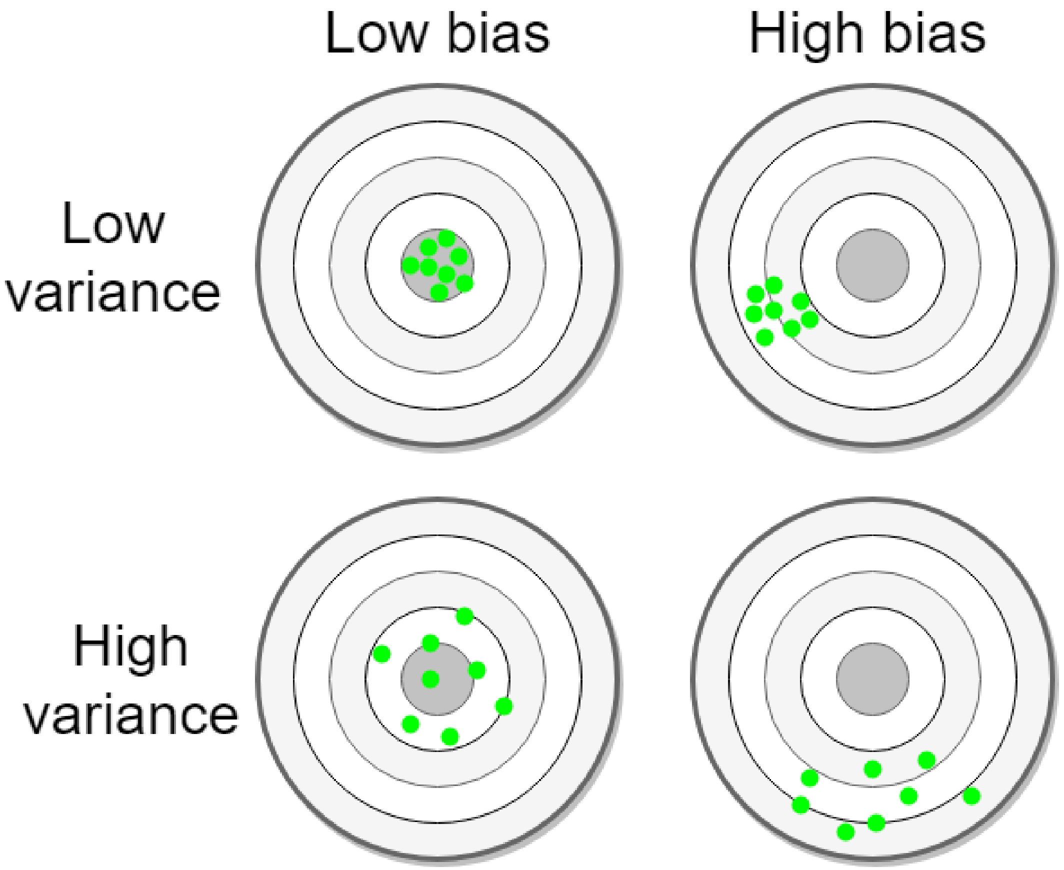
- 1000 people: 10 are really interested in, 900 are not interested at all
- Accuracy=(909+9)/1000=91.8%
- Sensitivity = 9/(9+1) = 90%
- Precision = 9/(9+81) = 10% ♣

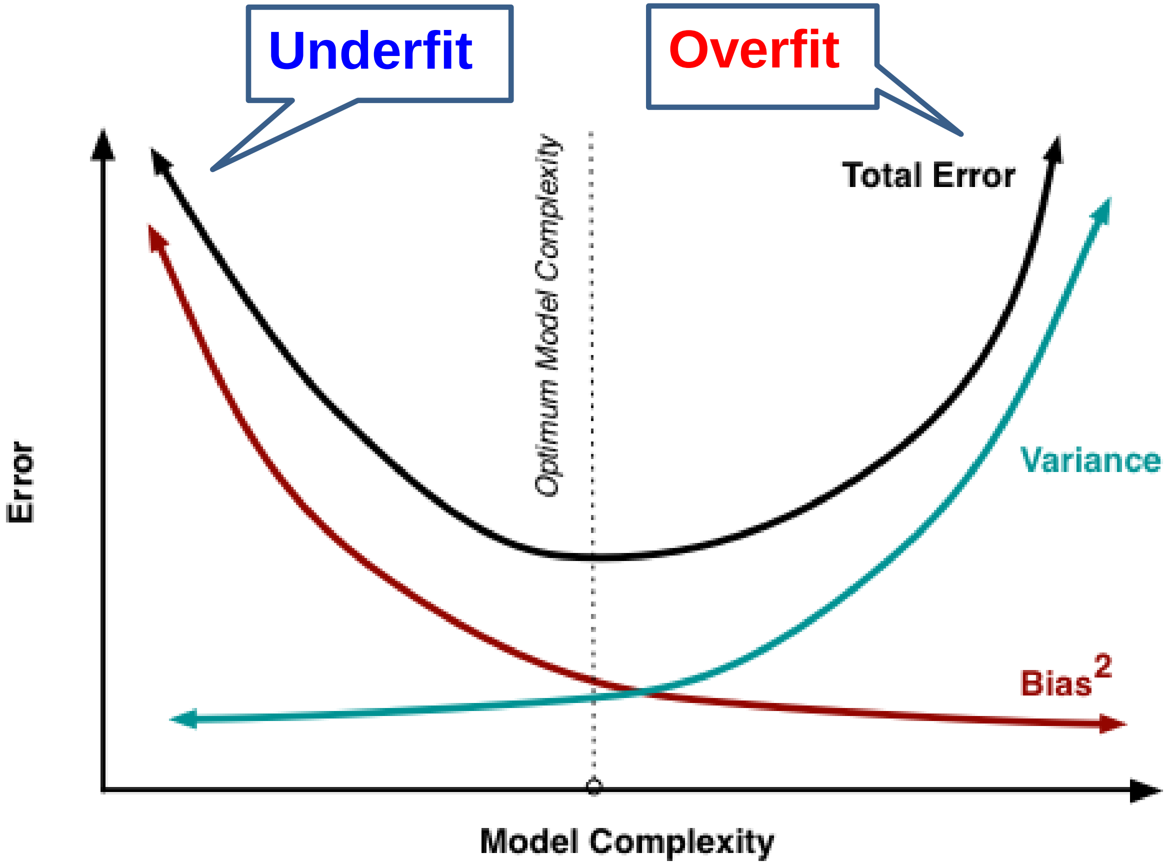
♥ **FP in advertising target wastes money !**

Overfit

Bias-Variance Tradeoff

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

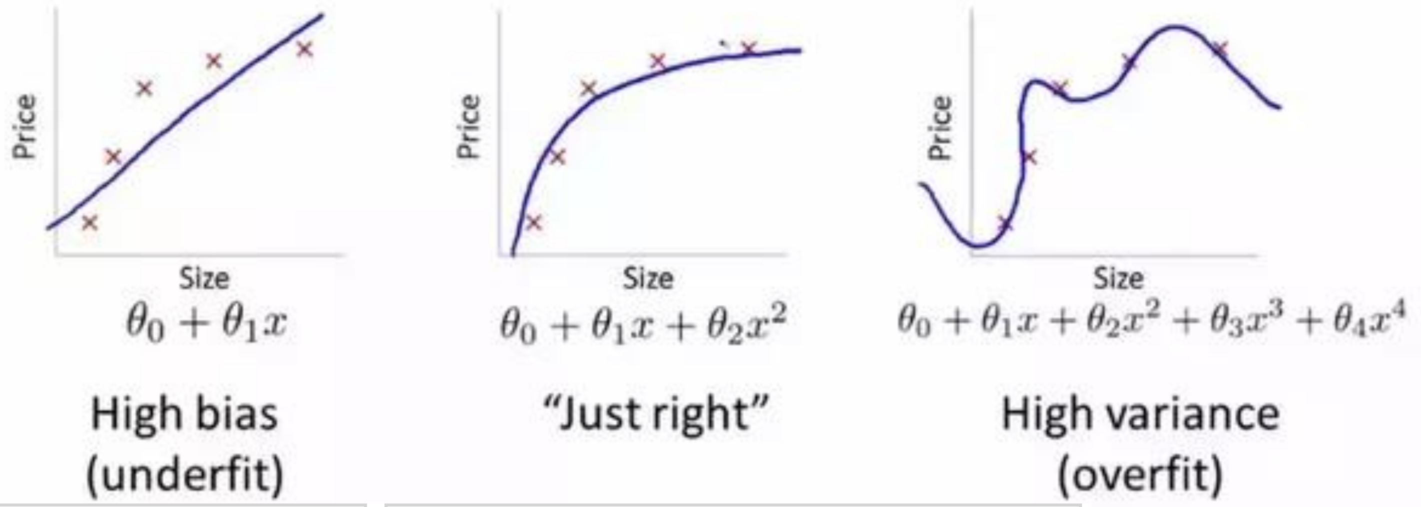




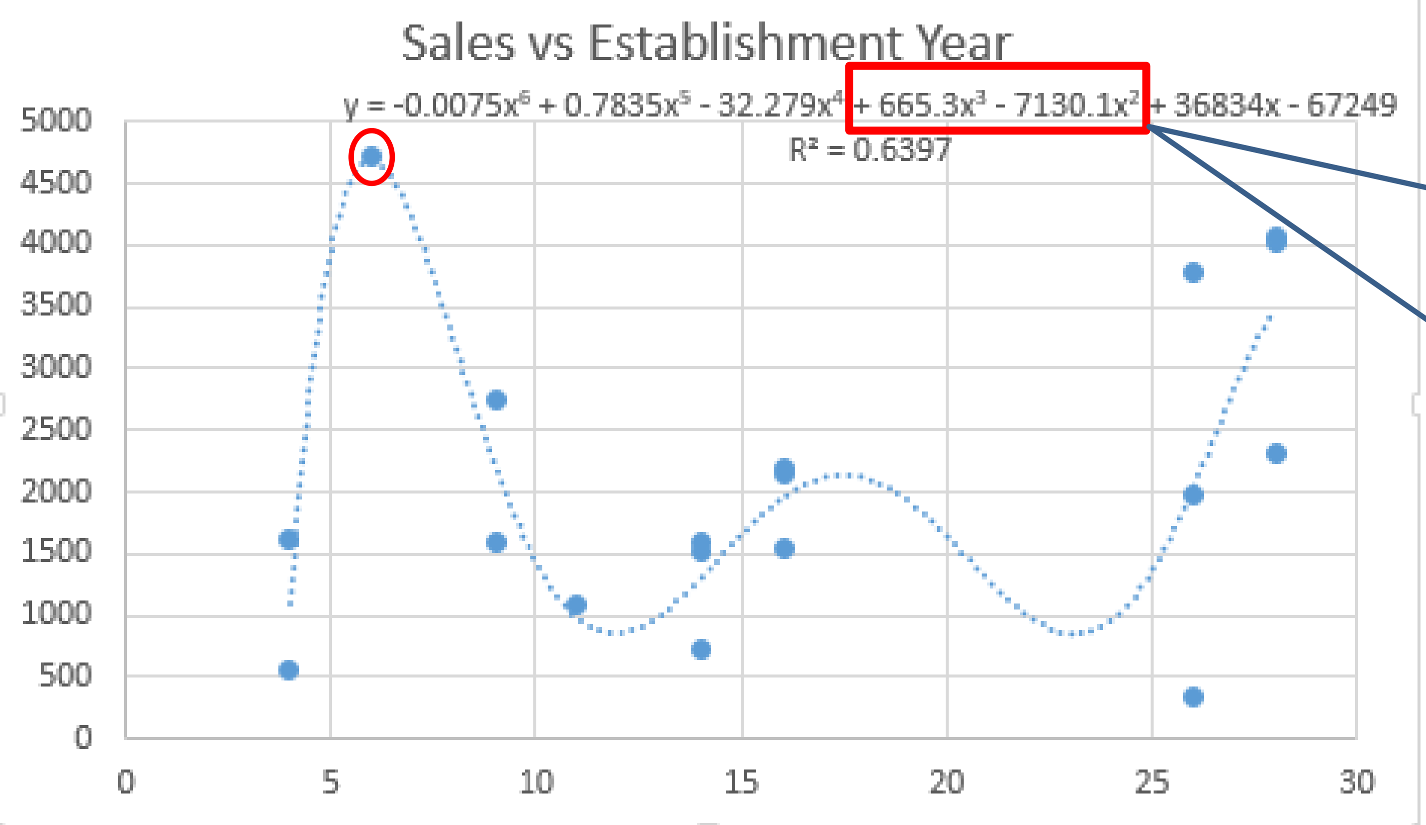
https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

Regression

- Problems:



- Example:

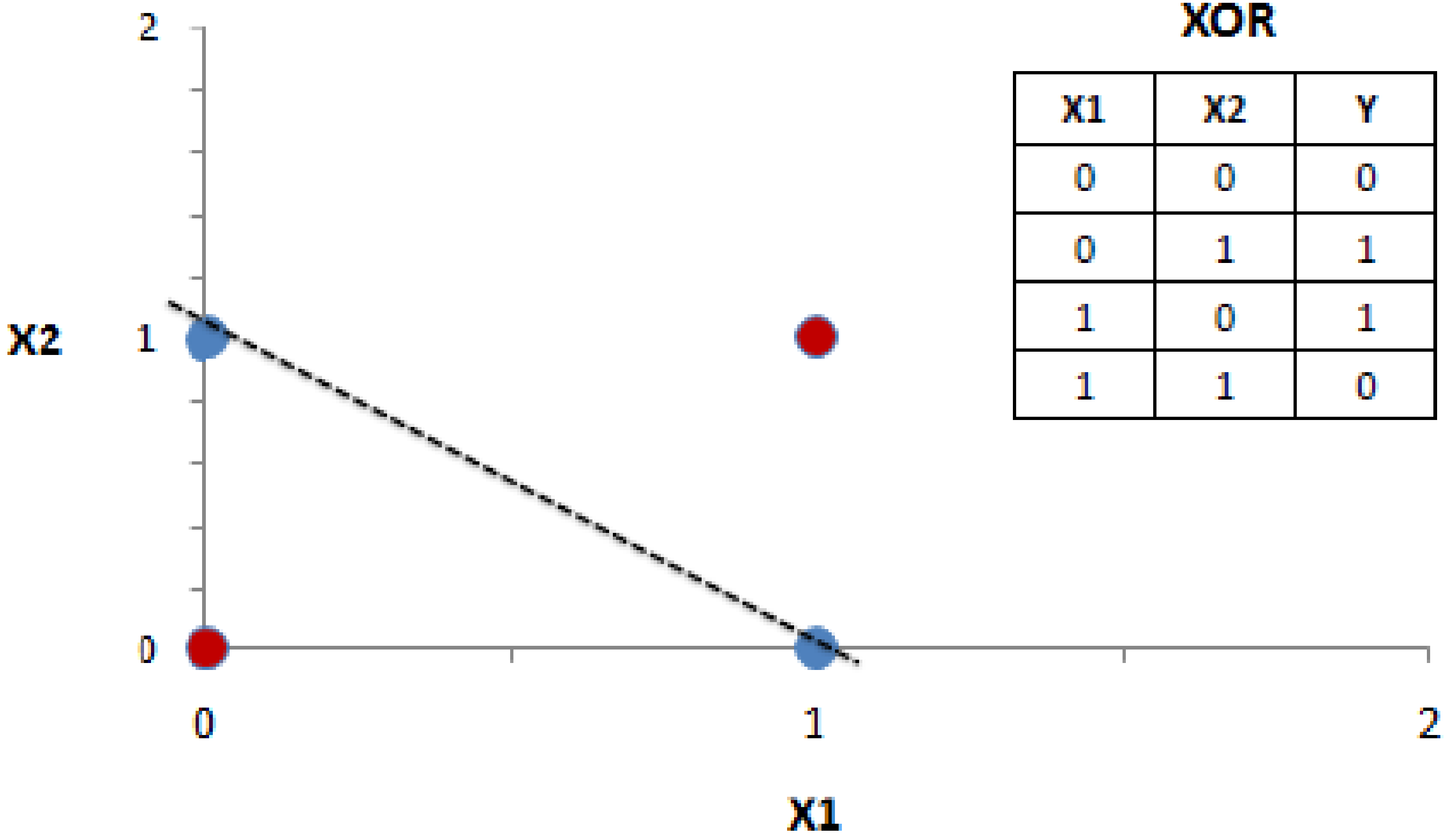


If you include more and more features/variables, your model will perform well on train data, but "overfit"

Regression

- Problems:

- Exclusive-OR:



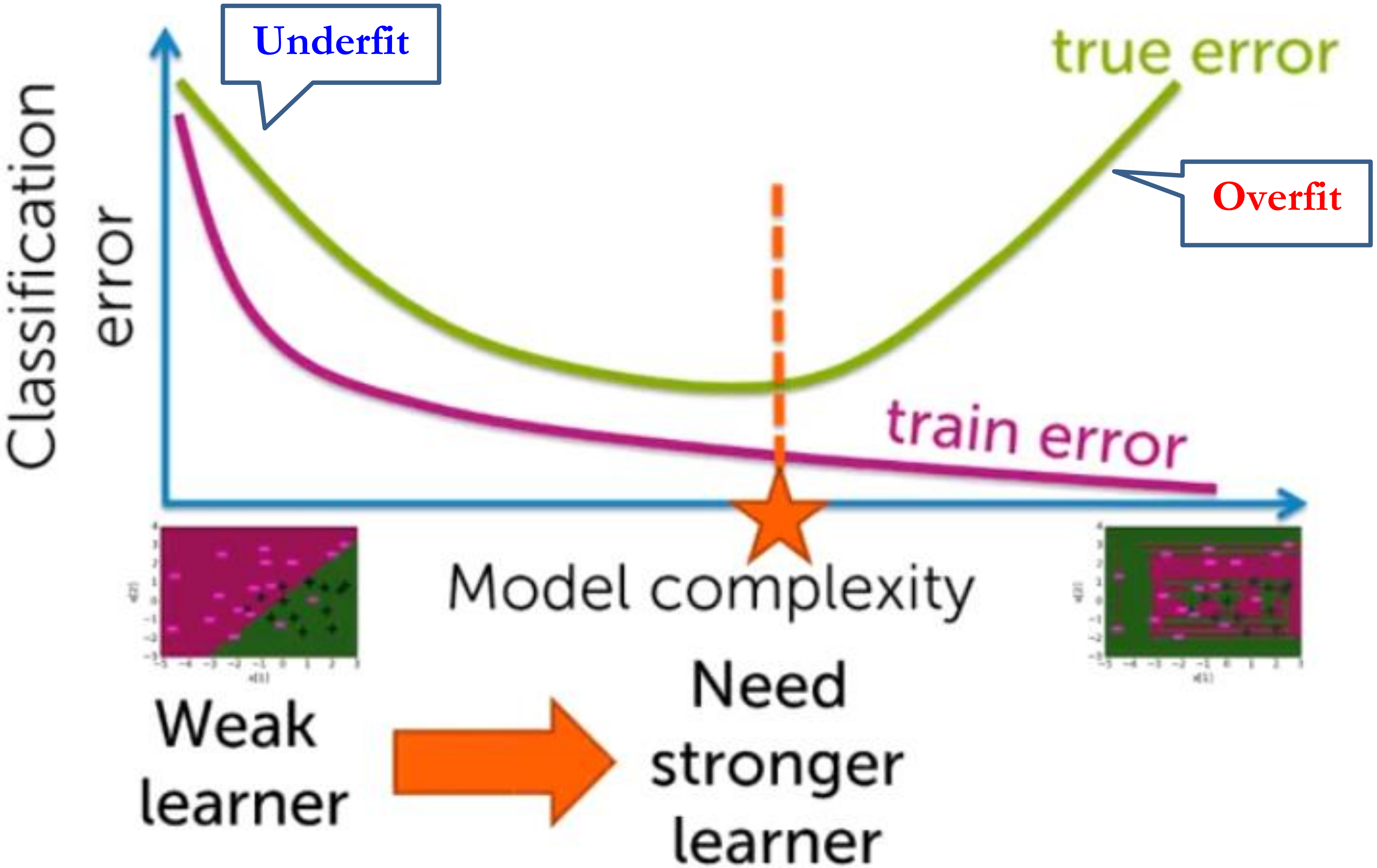
XOR

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Underfit And Overfit

- Finding a single model that performs well on prediction is not so easy.

- Data Scientists focus more on overfit issue.



Overfit

- How to control overfit issue:
    - Cross-Validation
    - Feature Selection
    - Pruning in decision tree (e.g., early stopping)
    - Ensemble Learning methods (e.g., random forest, boosting)
    - Dropout in deep learning and neural network
    - More…

Feature Selection

Feature Selection Methods:

- Brute-Force
  - Use simple logic or domain knowledge
- Filter method
  - correlations between feature X and output y
  - chi-square test, etc.
- Wrapper method
  - Sequential Feature Selection (e.g., stepwise regression)
- Embedded method
  - decision tree

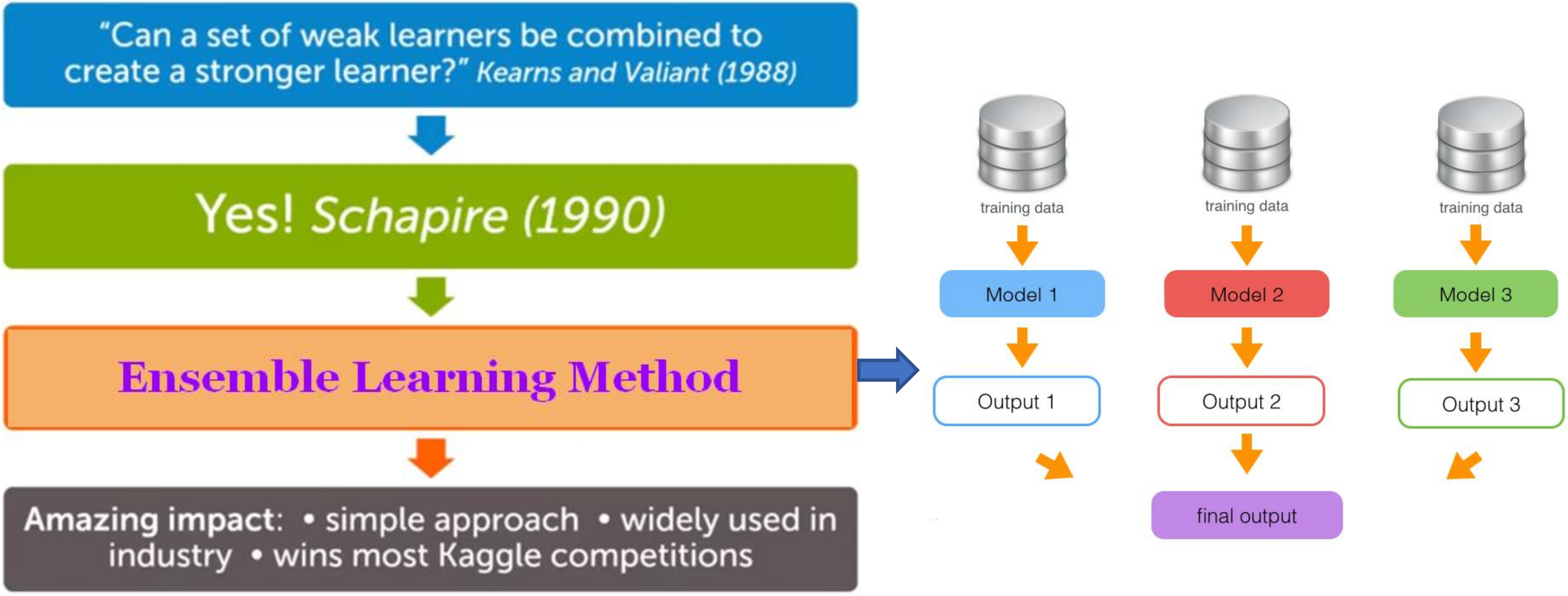
Cross Validation

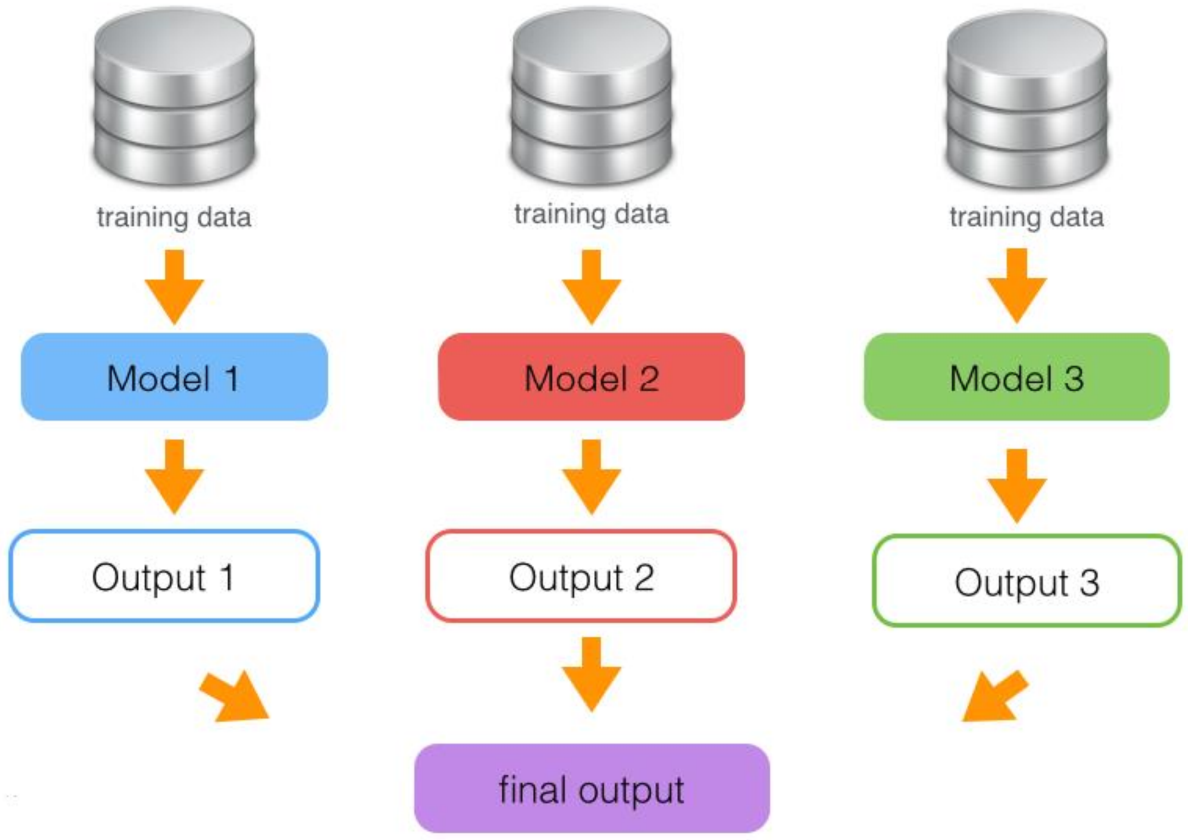
- K-Fold Cross Validation (e.g., K=10)

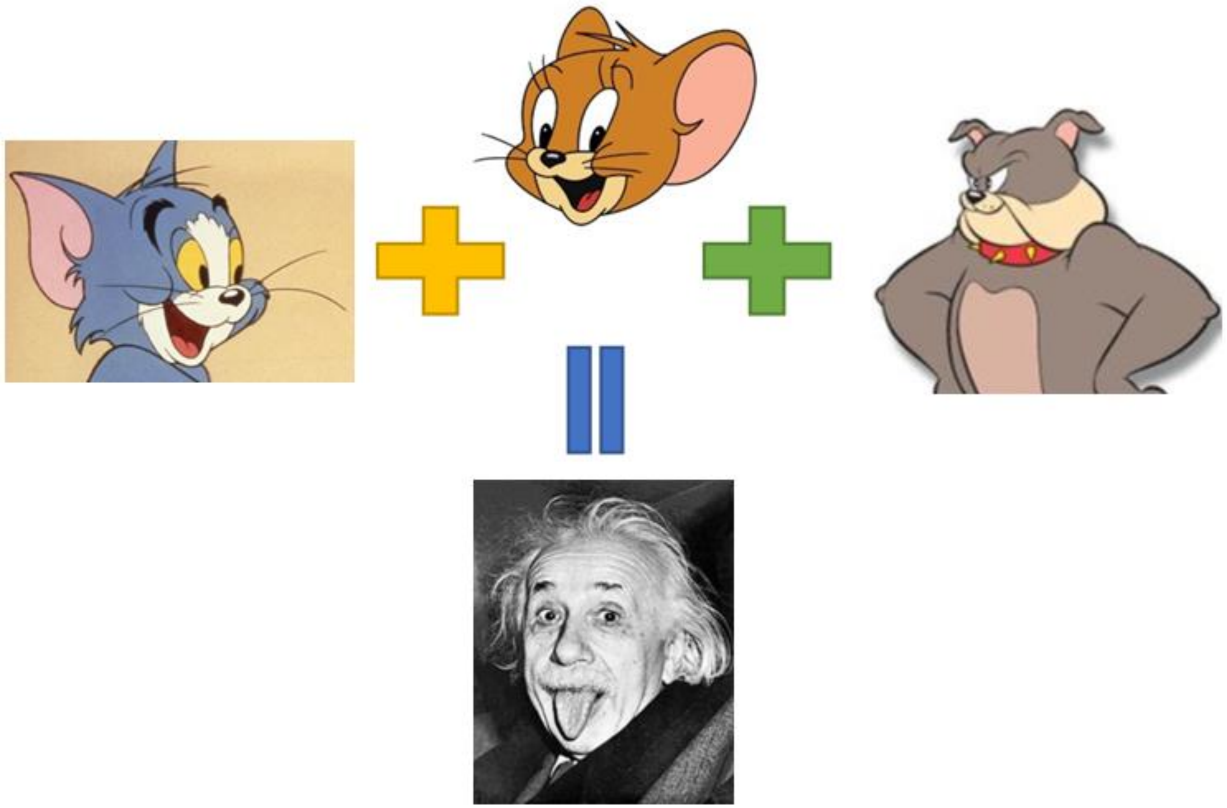
- Model Evaluation
- Model Comparison
- **Model Tuning**

K = 3: ≈ 70/30 split;
K = 5: = 80/20 split;
K = 10: = 90/10 split

Ensemble learning







Learn from not just one learner/model but a set of base learners/models, and **combine** their predictions for the unseen instances using some **aggregation methods** (e.g., taking average, majority voting, logistic regression, etc.)

A set of weak models are combined to create a strong model

Ensemble Method

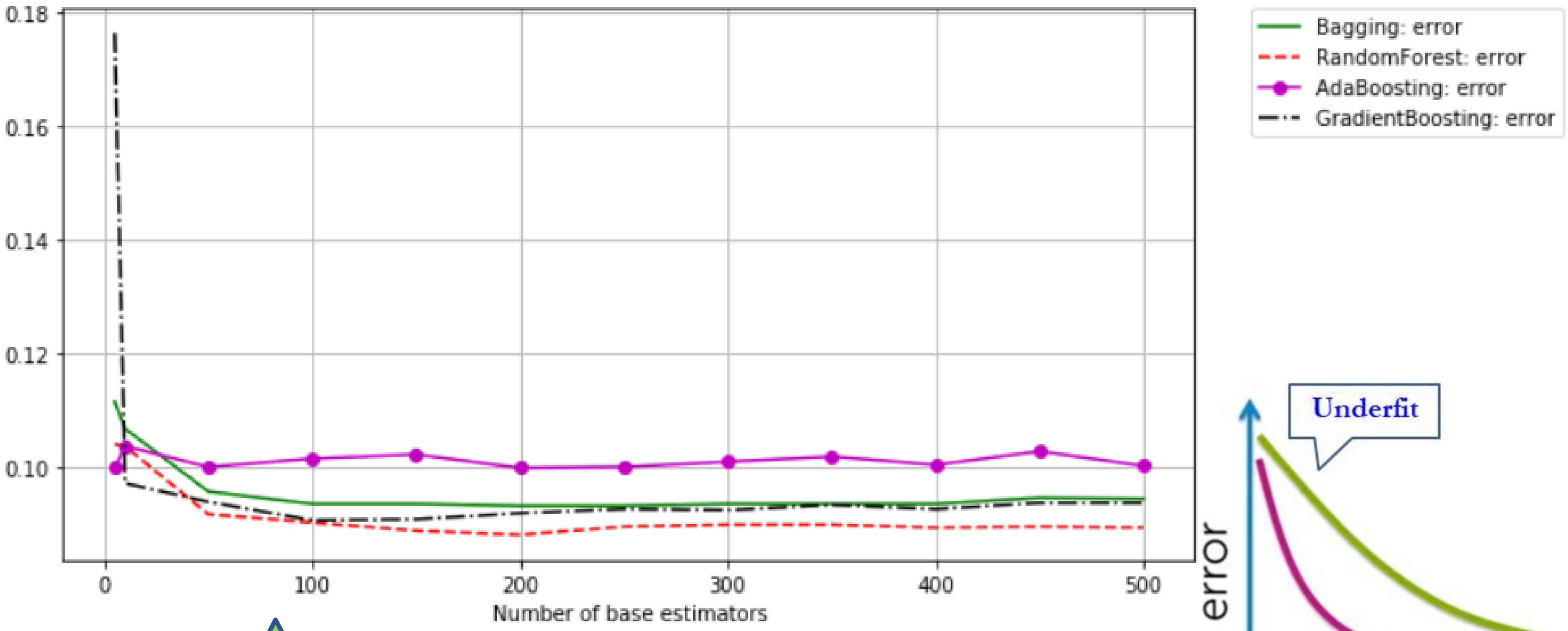
Parallel Learning

- Bagging (Bootstrapping Aggregation)
- Random Forest
- Stacking: Combination of different-type base models

Sequential Learning (Error-Based or Residual-Based)

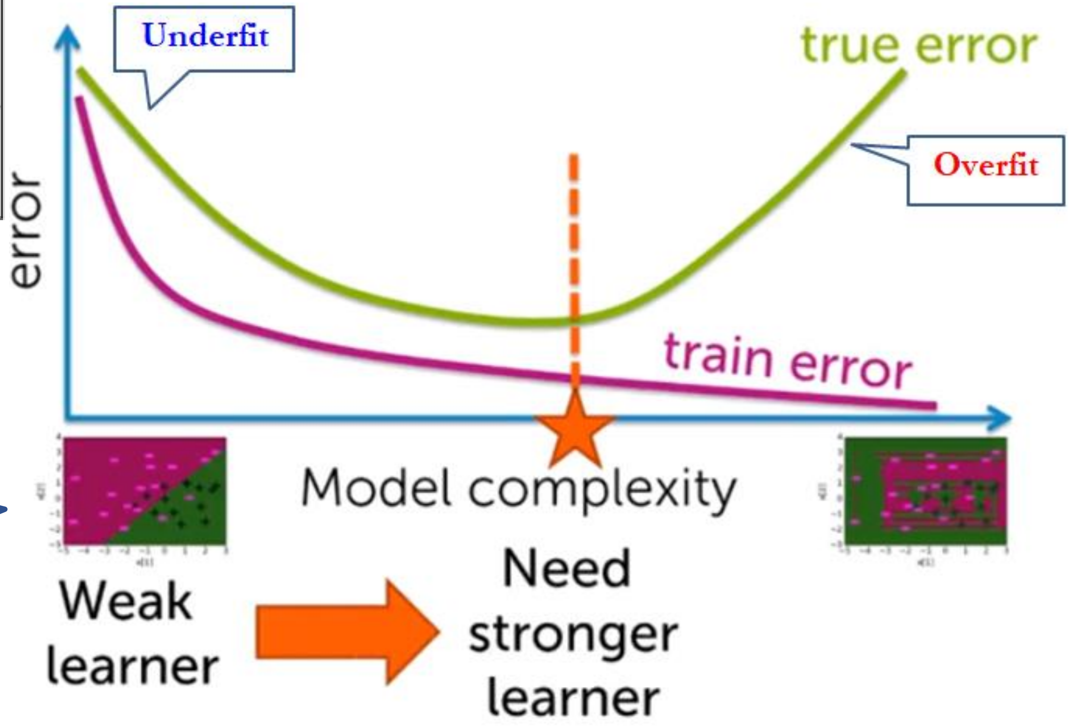
- Adaboost (Adaptive Boosting): Increase weights on misclassified data
- Gradient Boosting: Fit base models on residuals
- XGBoost: An extension from Gradient Boosting

Ensemble Method



Ensemble Learning

Single Model



Thank You !