الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

AI for Interdisciplinary Research

Part 1: AI as a Black Box – Concepts and Foundations

Dr. Riad Sonbol

April 2026

Workshop Objectives

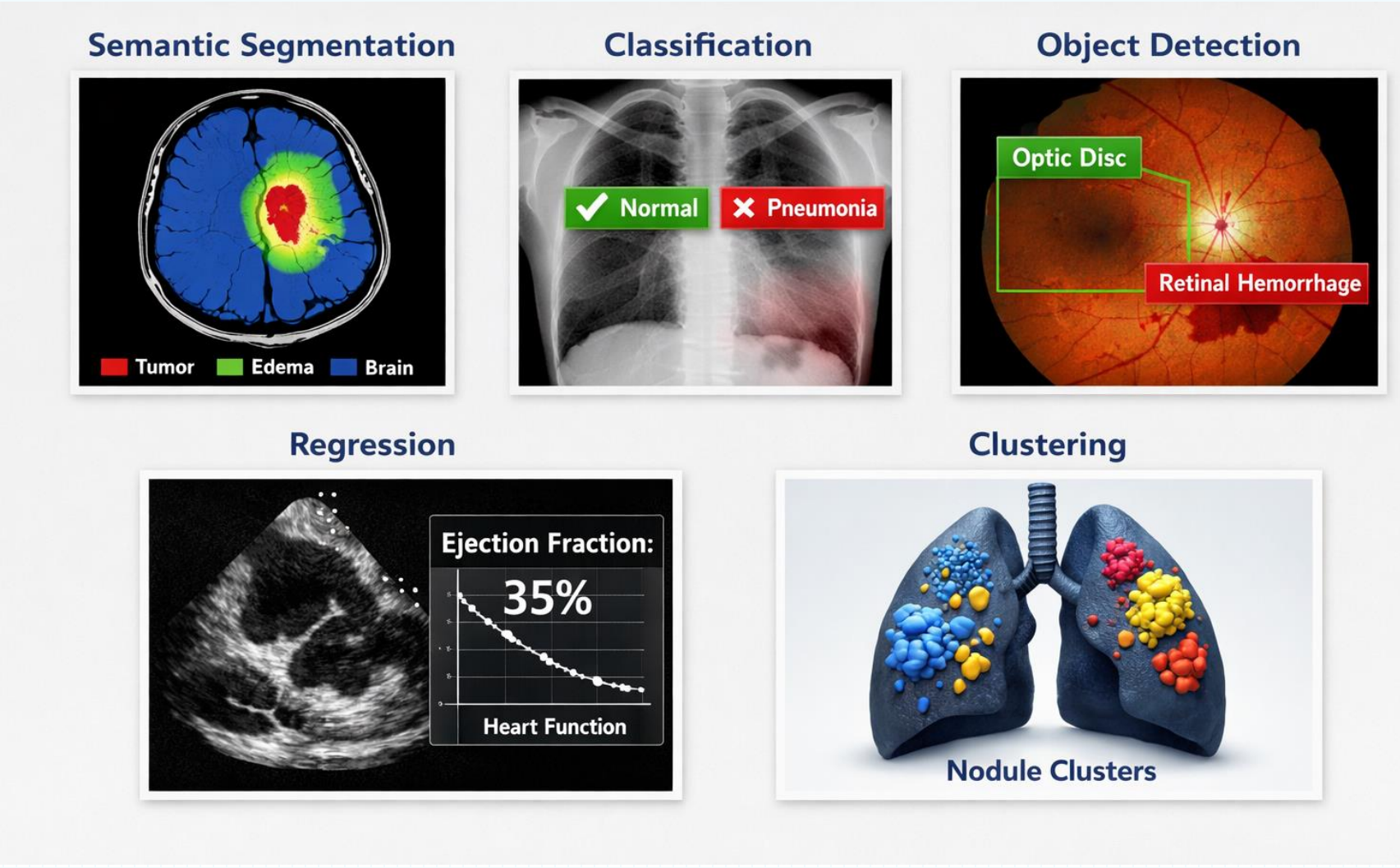
- Introduce artificial intelligence (AI) to researchers from **non-AI backgrounds.**
- Develop a clear understanding of **how AI methods work** and are applied across domains.
- Enable participants to critically **evaluate AI-based research papers** in their fields.
- Support effective **collaboration with AI specialists in interdisciplinary research projects.**

Road Map

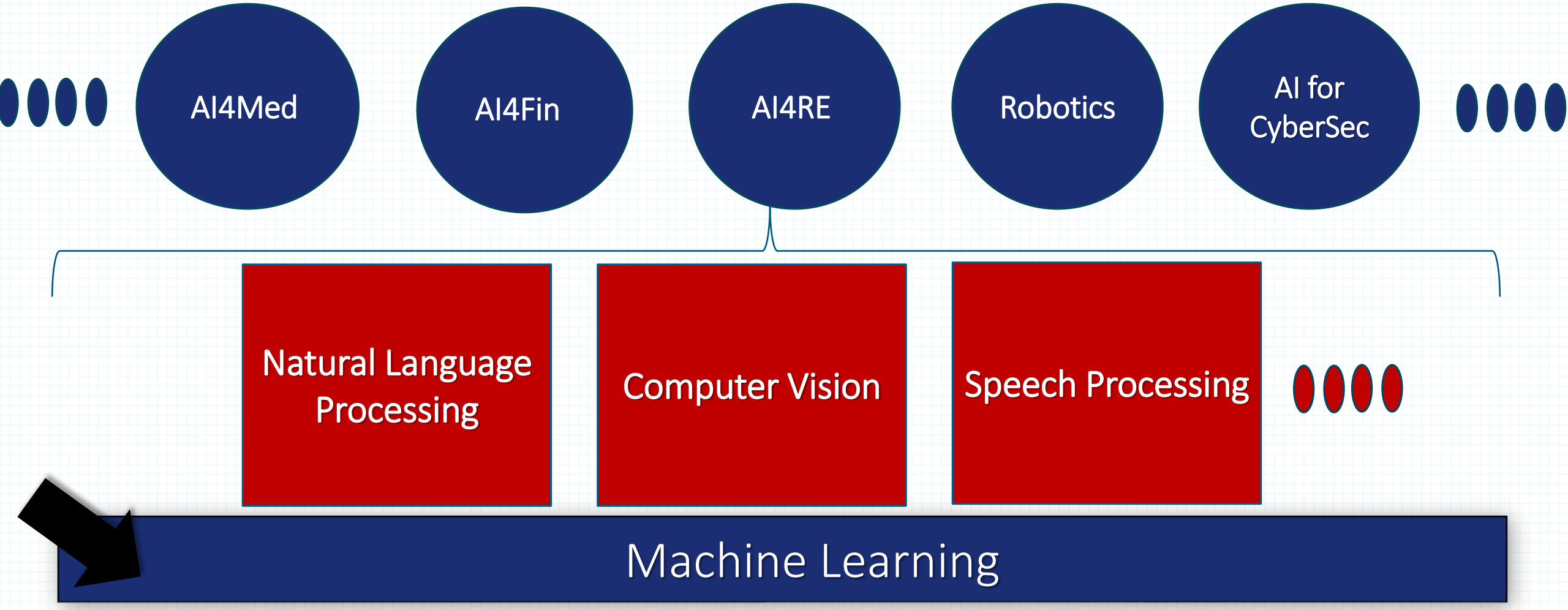
- AI as a Black Box – Concepts and Foundations
- Opening the Box (1) – Traditional Machine Learning
- Opening the Box (2) – Deep Learning Concepts
- Modern AI and Research Directions

What is artificial intelligence?

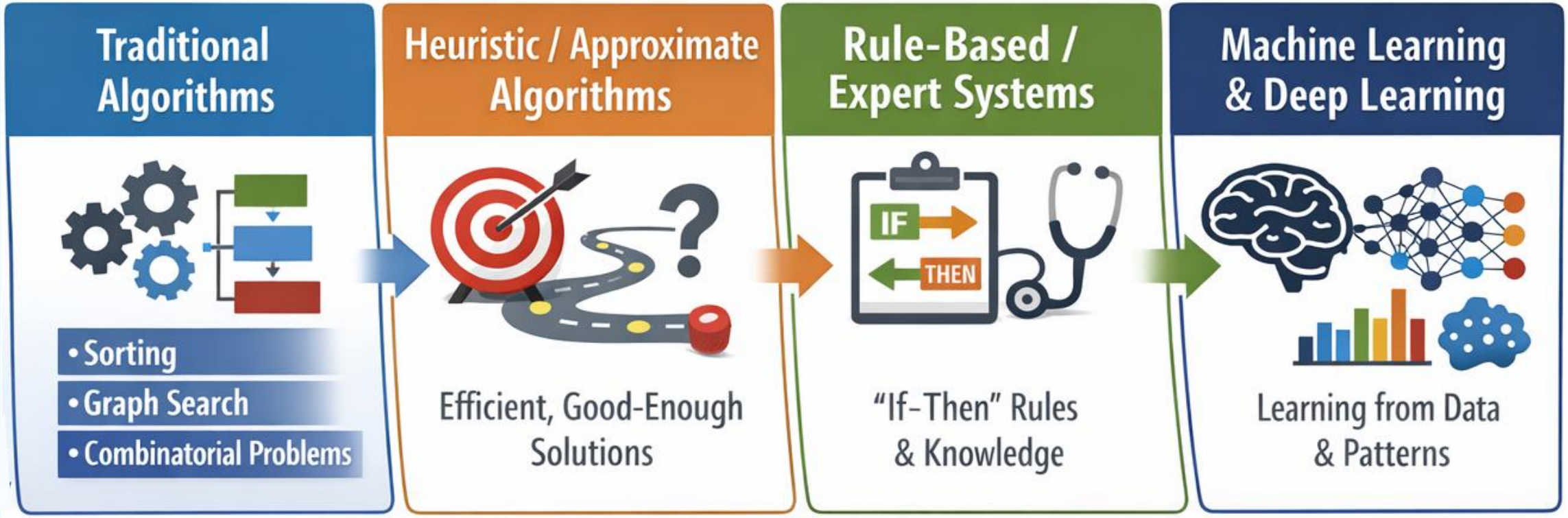
- Artificial intelligence is computer software capable of performing activities that **resemble human behavior**, such as:
  - Learning from data,
  - Reasoning and making decisions,
  - Self-correcting errors



“Modern” AI Researches



AI as a Problem-Solving Paradigm



exact steps to solve clear problems

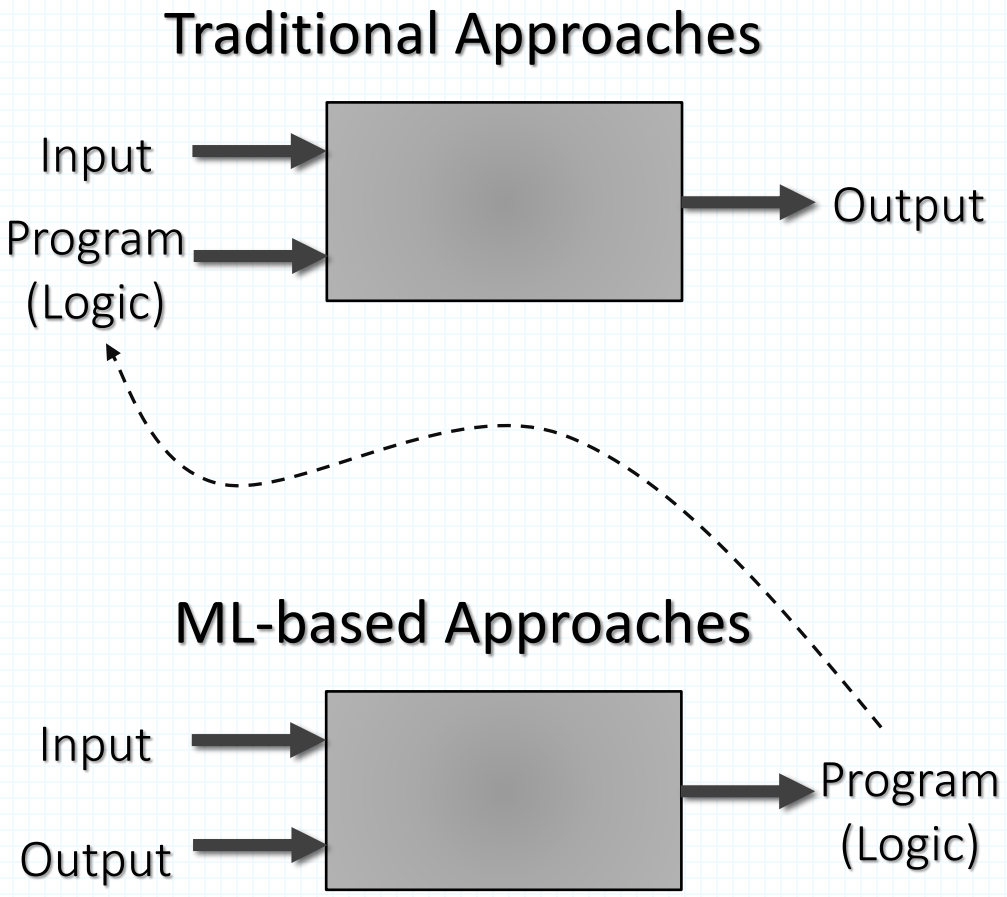
find Approximate solutions quickly when exact ones are too slow

use human "if-then" rules for specific tasks

learn patterns from data

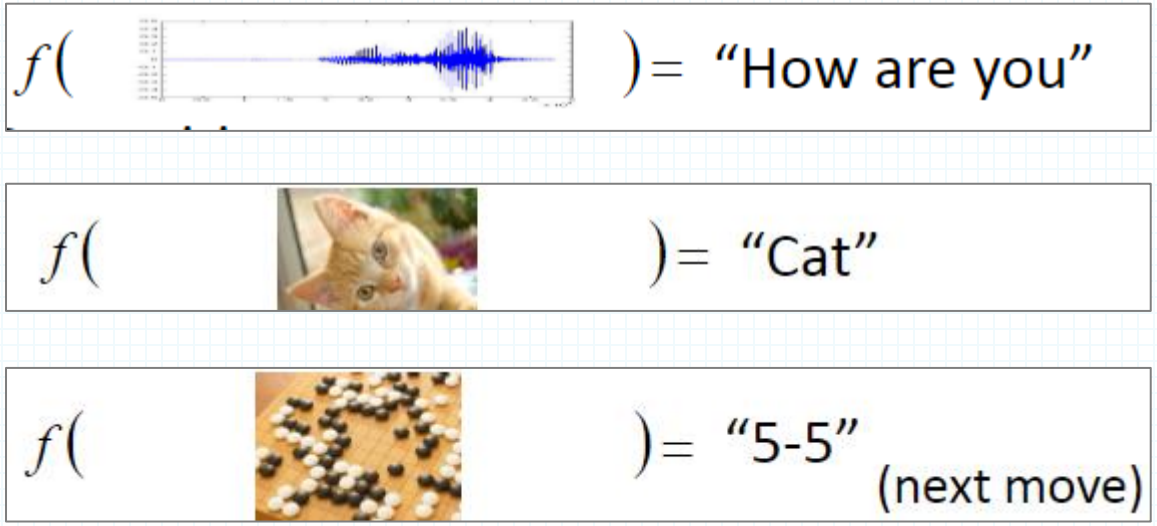
What is Machine Learning?

- Machine Learning is a type of Artificial Intelligence that provides computers with the ability to learn
- Getting computers to program themselves.



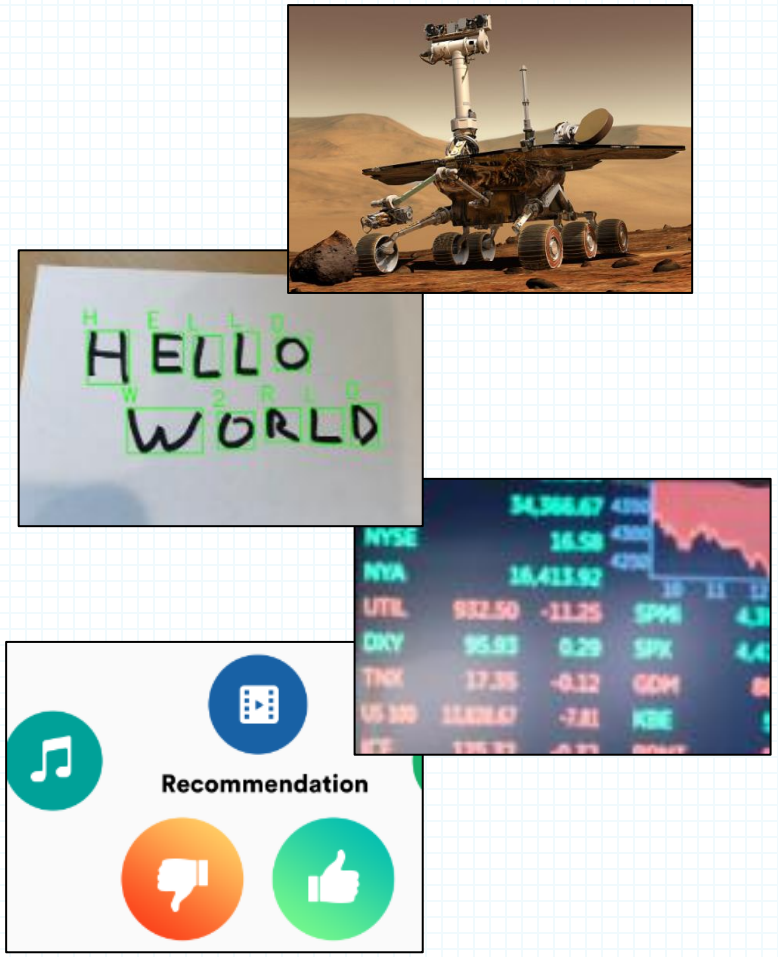
Machine Learning ≈ Looking for a Function

- At its core, **Machine Learning is about finding a function** that maps inputs to outputs:
  - Where x is your input (features, observations)
  - y is the target (label, prediction)
  - And f is the function learned from data
- Example:



When Do We Use Machine Learning?

- **ML is used when:**
  - Human **expertise does not exist** (navigating on Mars),
  - Humans are **unable to explain** their expertise (speech recognition, OCR)
  - Solution **changes in time** (routing on a computer network, stock market)
  - Solution needs to be **adapted to particular cases** (recommendation systems)
- **Learning is not always useful:**
  - There is no need to “learn” to calculate payroll.

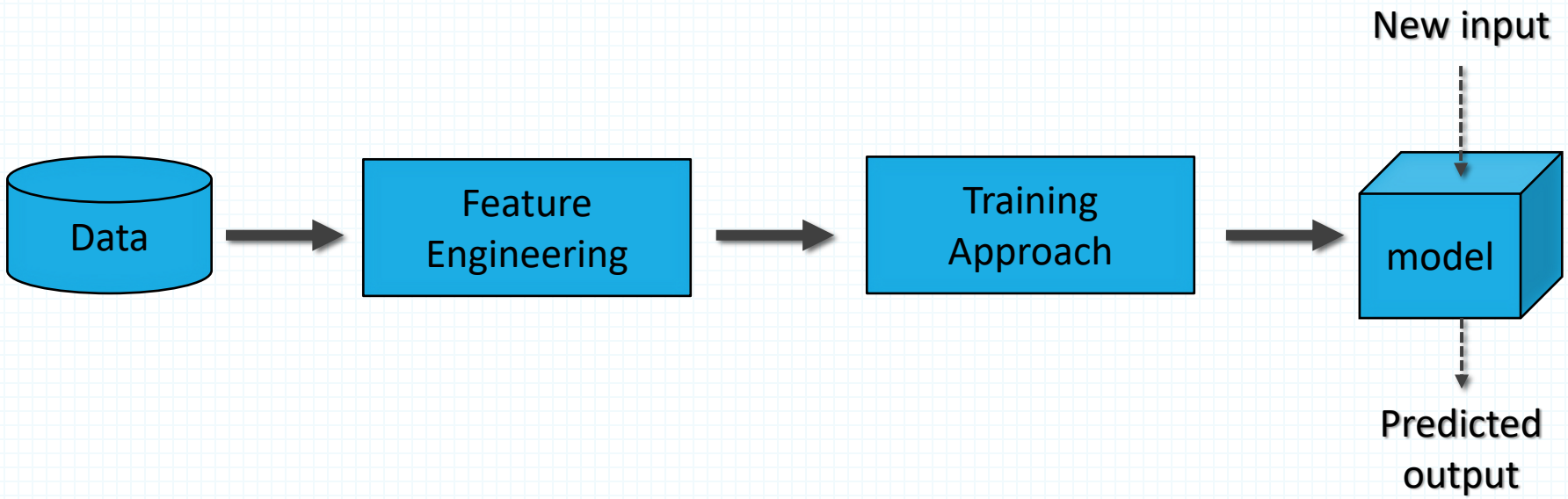


Types of Learning

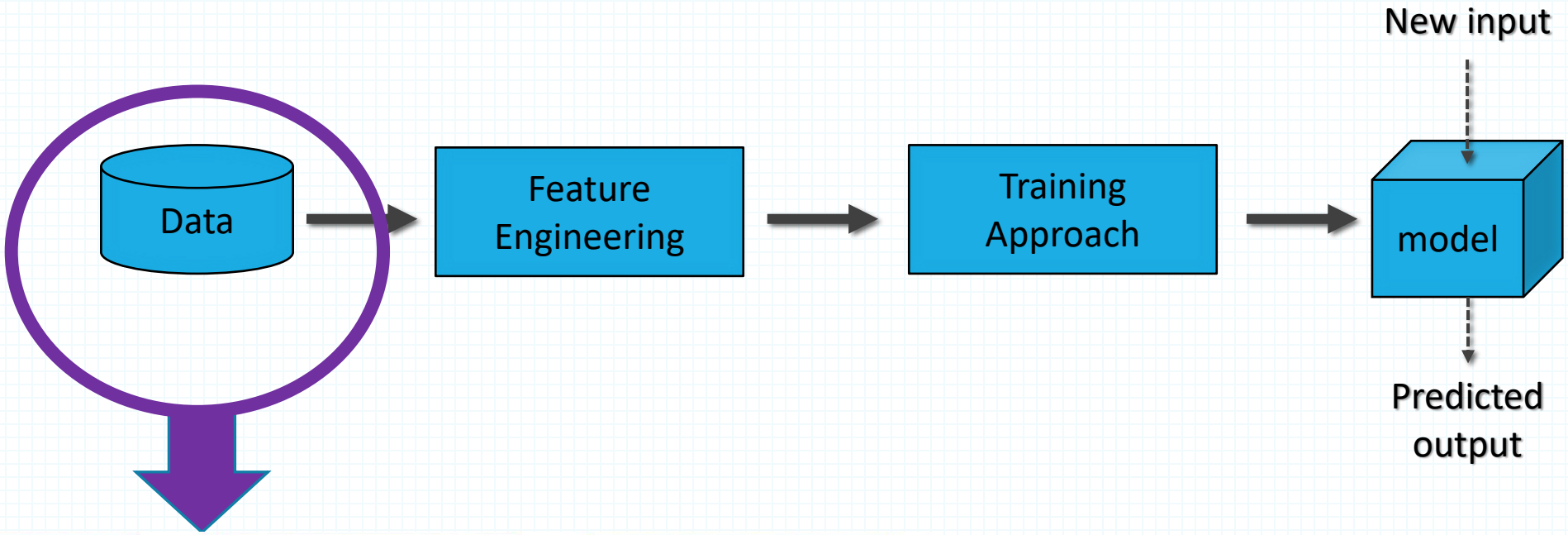
- **Supervised (inductive) learning:** Training data includes desired outputs
  - Regression: predict numerical values
  - Classification: predict categorical values, i.e., labels
- **Unsupervised learning:** Training data does not include desired outputs
  - Clustering: group data according to "distance"
  - Association: find frequent co-occurrences
- **Reinforcement learning**
  - Learn to act based on feedback/reward.
- **Self-Supervised Learning.**
- etc

ML Pipeline

Traditional ML Pipeline



Traditional ML Pipeline



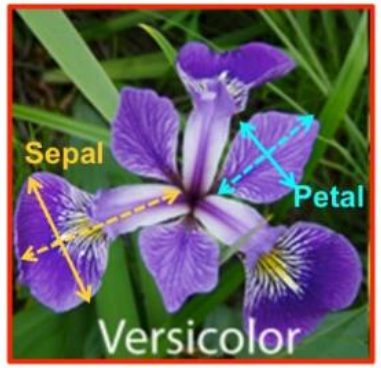
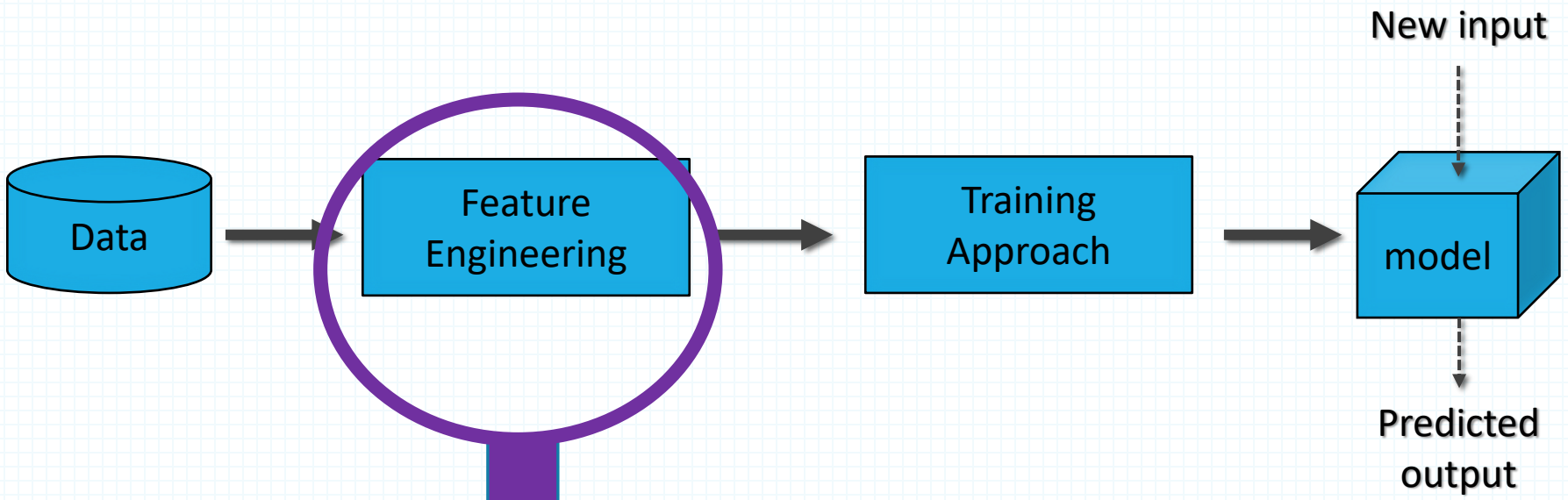


Image Source: http://suruchifialoke.com/2016-10-13-machine-learning-tutorial-iris-classification/

Traditional ML Pipeline



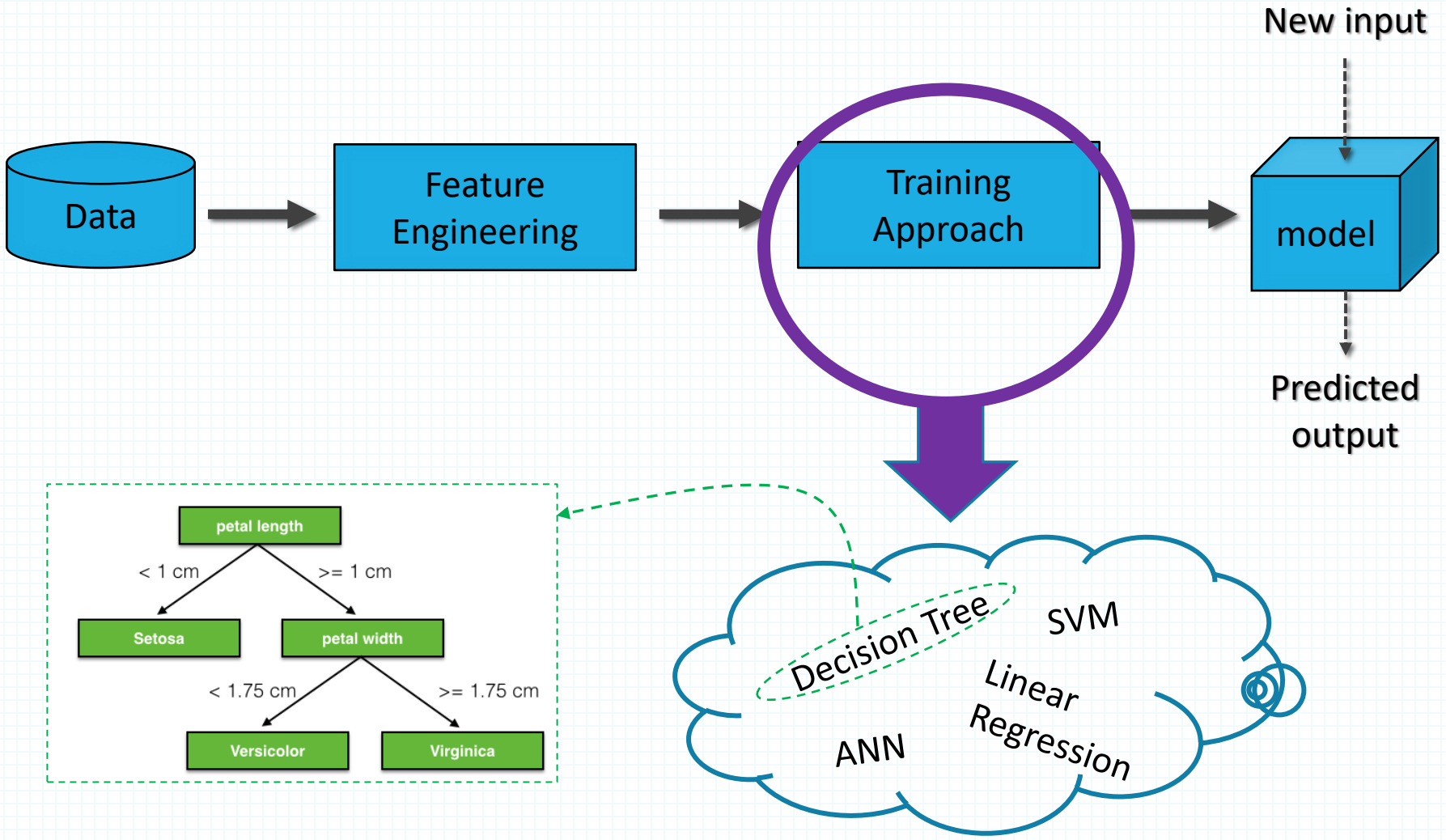
Features

Class (Output)

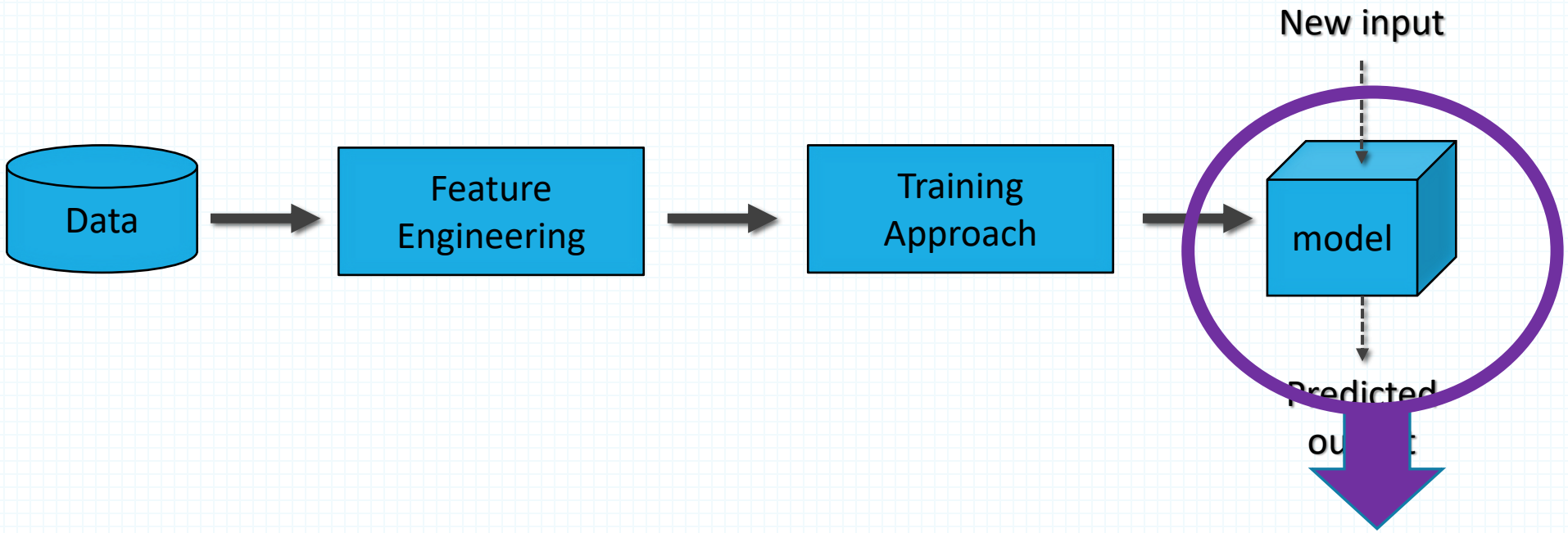
```
1 sepal_length,sepal_width,petal_length, petal width,class
2 5.1,3.5,1.4,0.2,Iris-setosa
3 4.9,3.0,1.4,0.2,Iris-setosa
4 4.7,3.2,1.3,0.2,Iris-setosa
5 4.6,3.1,1.5,0.2,Iris-setosa
6 5.0,3.6,1.4,0.2,Iris-setosa
7 5.4,3.9,1.7,0.4,Iris-setosa
8 4.6,3.4,1.4,0.3,Iris-setosa
```

Instance

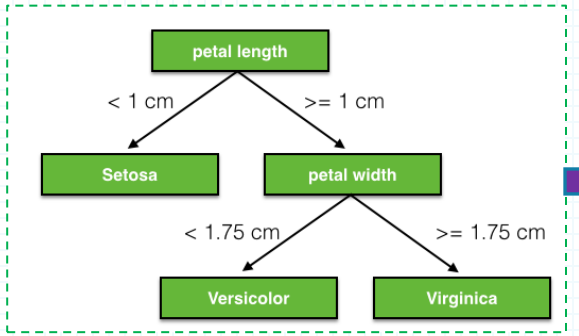
Traditional ML Pipeline



Traditional ML Pipeline

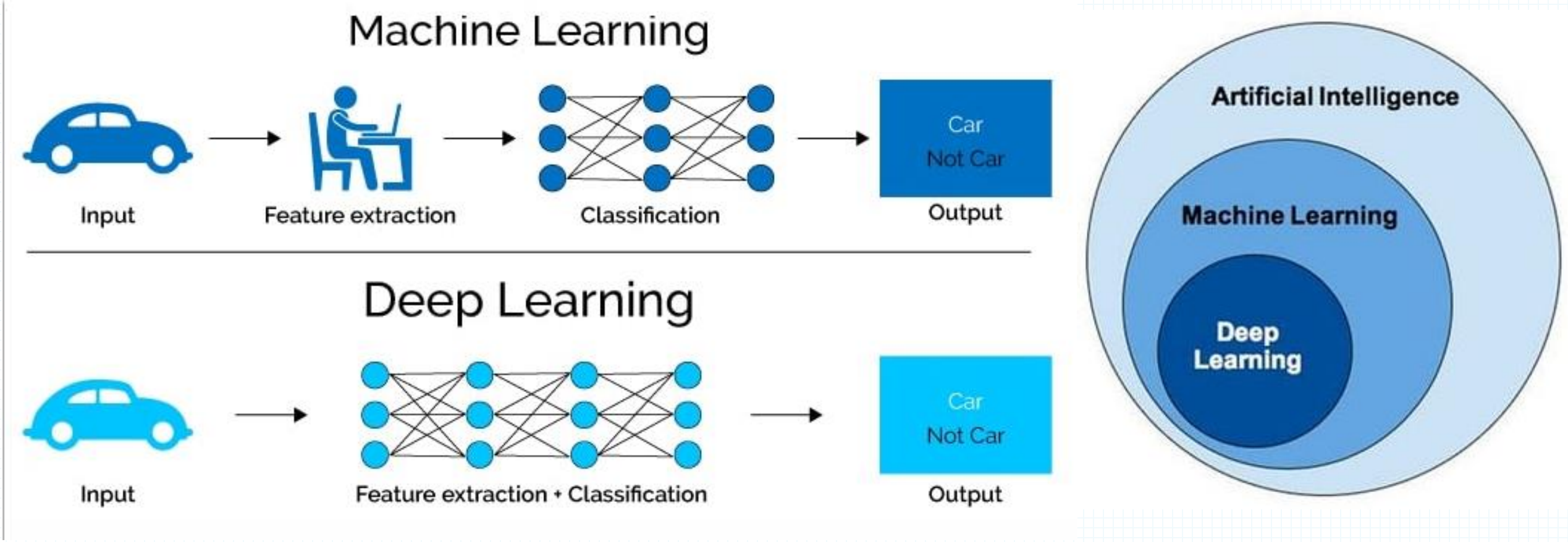


Sepal Length: 5.1 ,
Sepal Width=3.5,
Petal Length = 0.8,
Petal Width = 0.2,
Class = ???



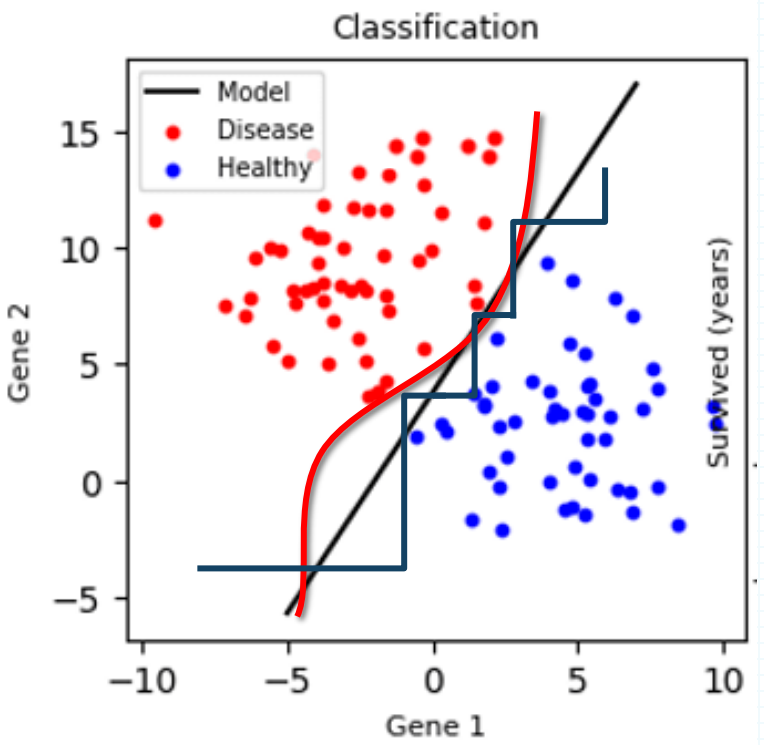
Class = Setosa

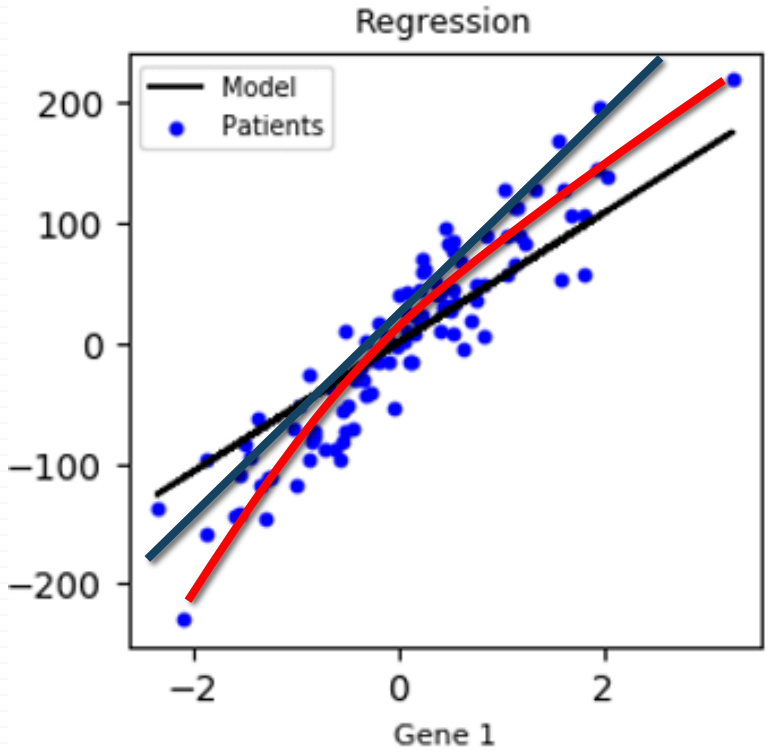
Traditional ML vs Deep Learning



Challenges in Machine Learning: Model Selection and Generalization

One of the main Challenge in ML:
The vast Number of Possible ML models!

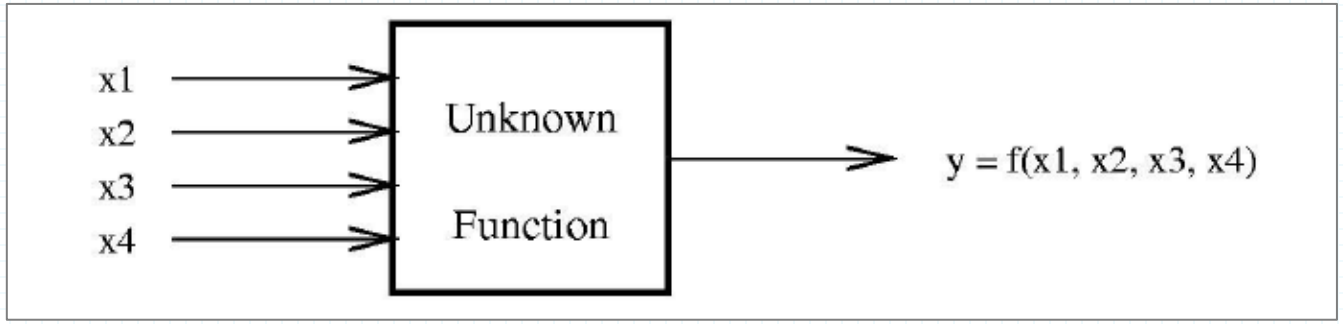




Which Dissension Boundaries is better!

One of the main Challenge in ML: The vast Number of Possible ML models!

Example



Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

What is the number of possible options?

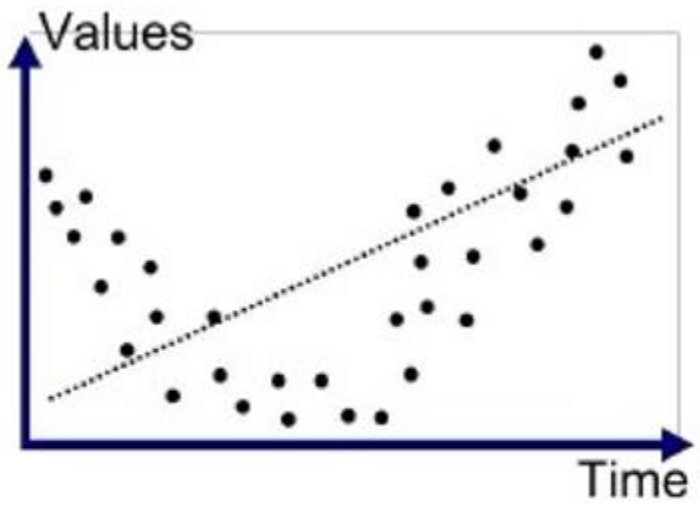
One of the main Challenge in ML: The vast Number of Possible ML models!

- 4 Boolean features
  - 2x2x2x2=16 options
  - Number of possible functions: 2^{16}
- We know 7 examples:
  - Number of possible functions: 2^9

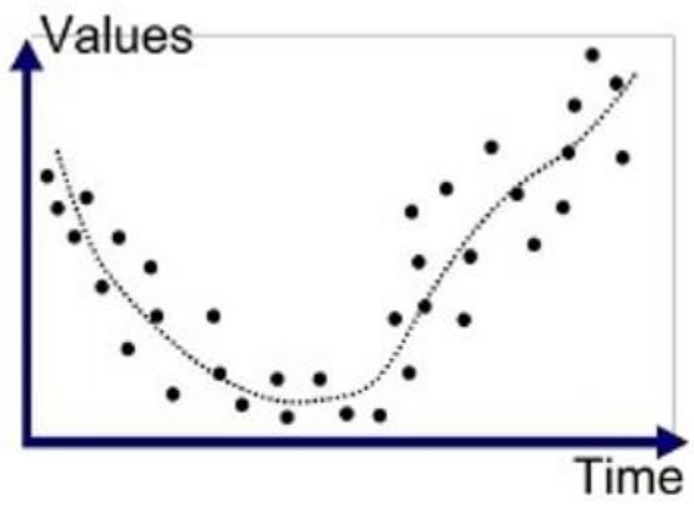
So.. Which Model is better?

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

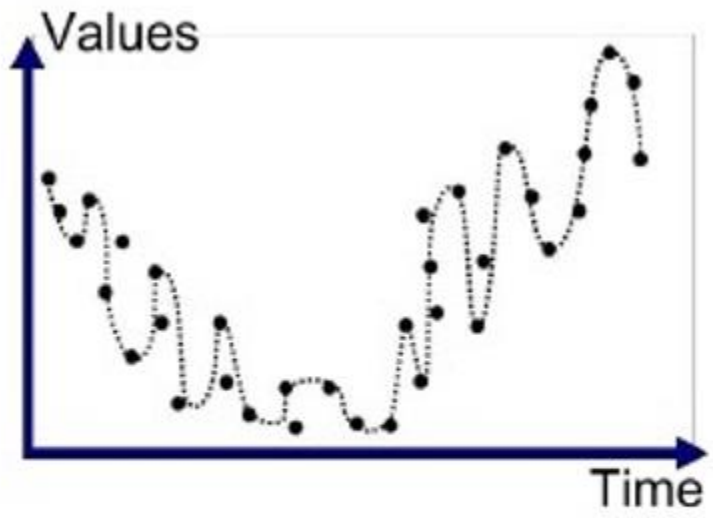
So.. Which Model is better?



Underfitted



Good Fit/Robust



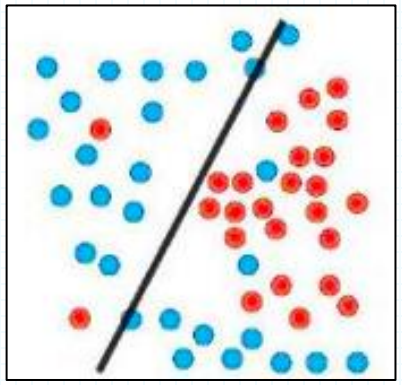
Overfitted

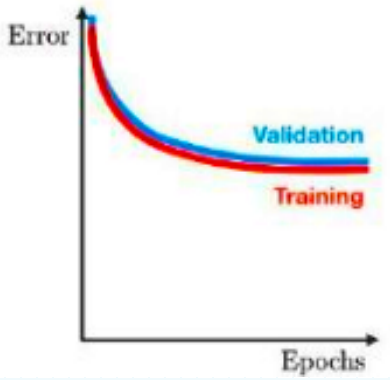
Generalization

"model's ability to adapt properly to new, previously unseen data"

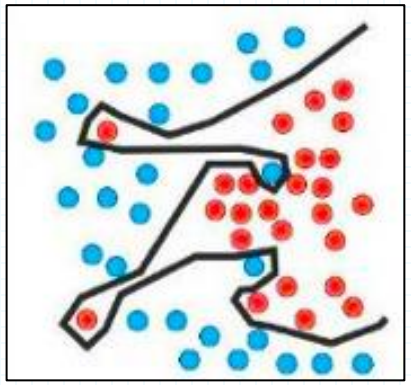
Overfitting vs. Underfitting

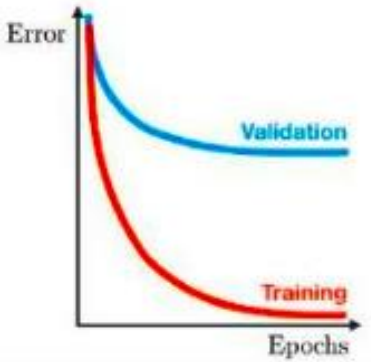
- ***Underfitting***
  - The model is too “simple” to represent all the relevant class characteristics
  - E.g., model with too few parameters produces high error on the training set and high error on the validation set





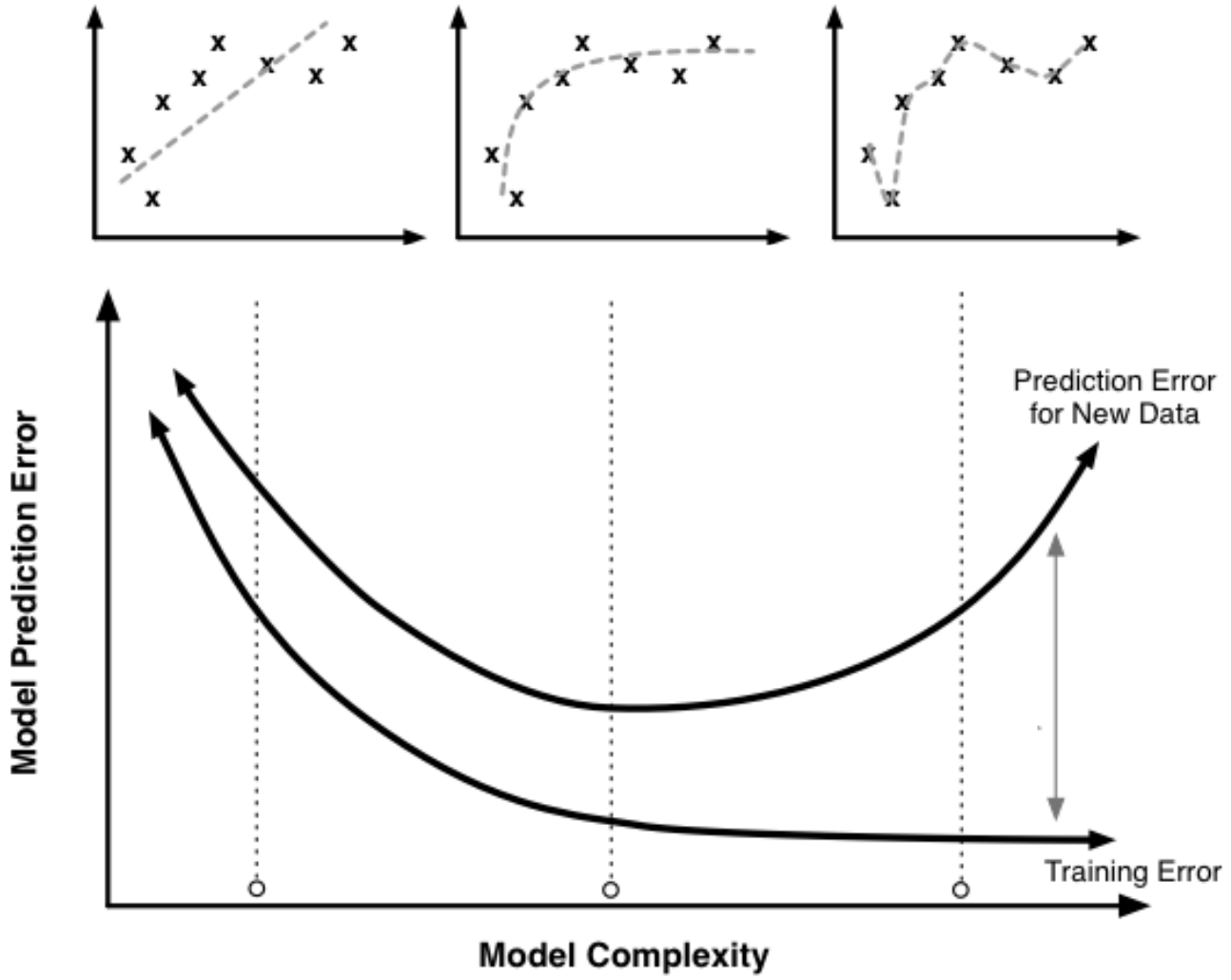
- ***Overfitting***
  - The model is too “complex” and fits irrelevant characteristics (noise) in the data
  - E.g., model with too many parameters produces low error on the training set and high error on the validation set





Overfitting vs. Underfitting

- **Overfitting:** The model learns noise and specific patterns in the training data, leading to poor performance on new data.
- **Underfitting:** The model is too simple and fails to capture meaningful patterns.



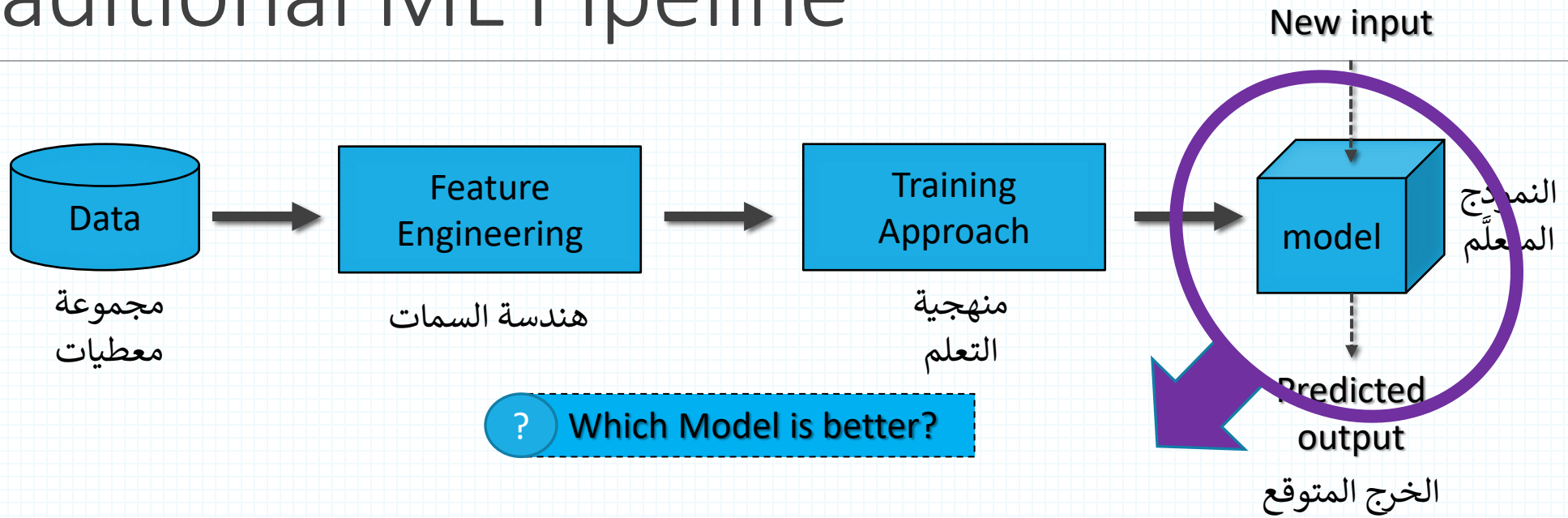
Generalization

Generalization

"model's ability to adapt properly to new, previously unseen data"

Estimation Strategy and Evaluation Metrics

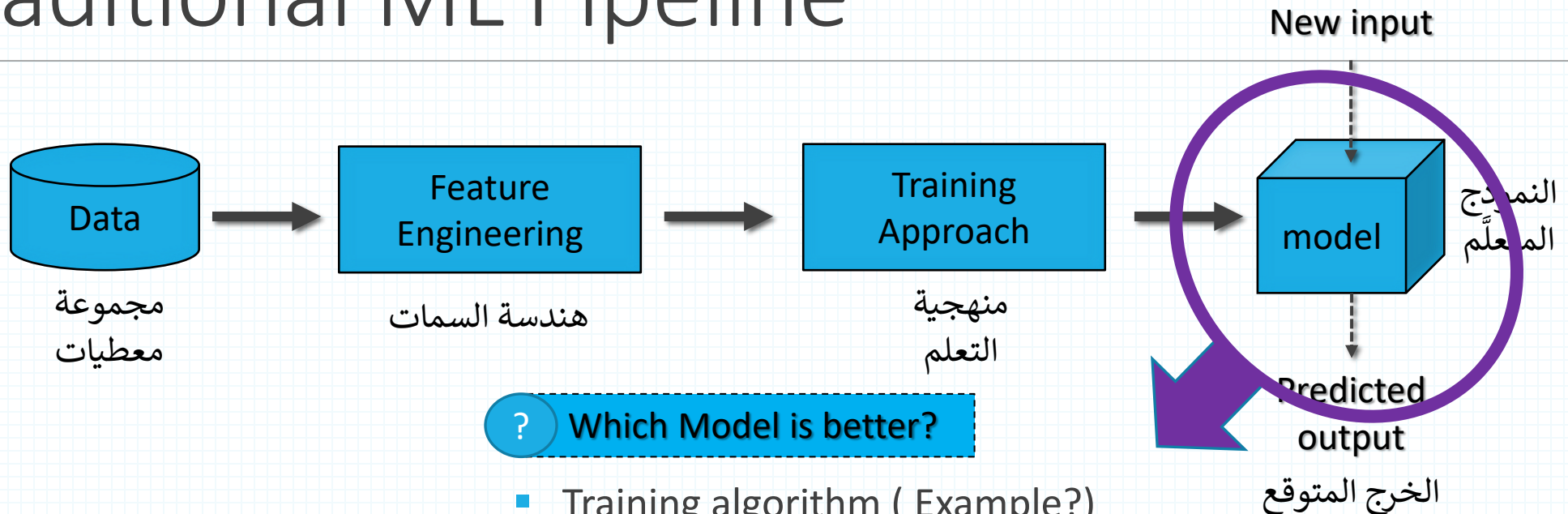
Traditional ML Pipeline



"All models are wrong; some are useful."

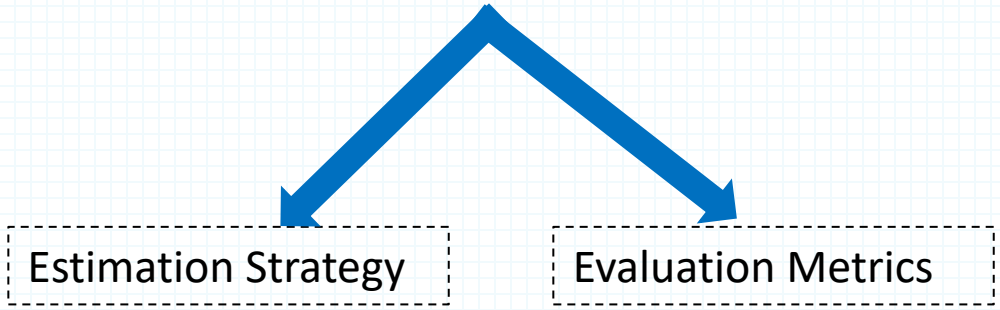
—George E. P. Box

Traditional ML Pipeline



? Which Model is better?

- Training algorithm (Example?)
- Parameters (Hyperparameters)?
- Handle unseen cases



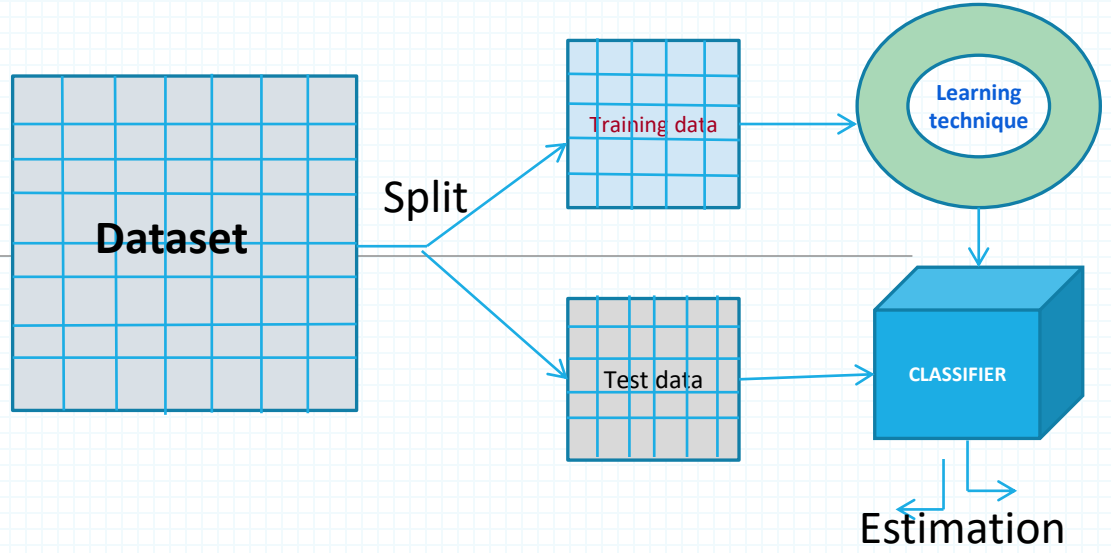
Estimation Strategy

Hyperparameters

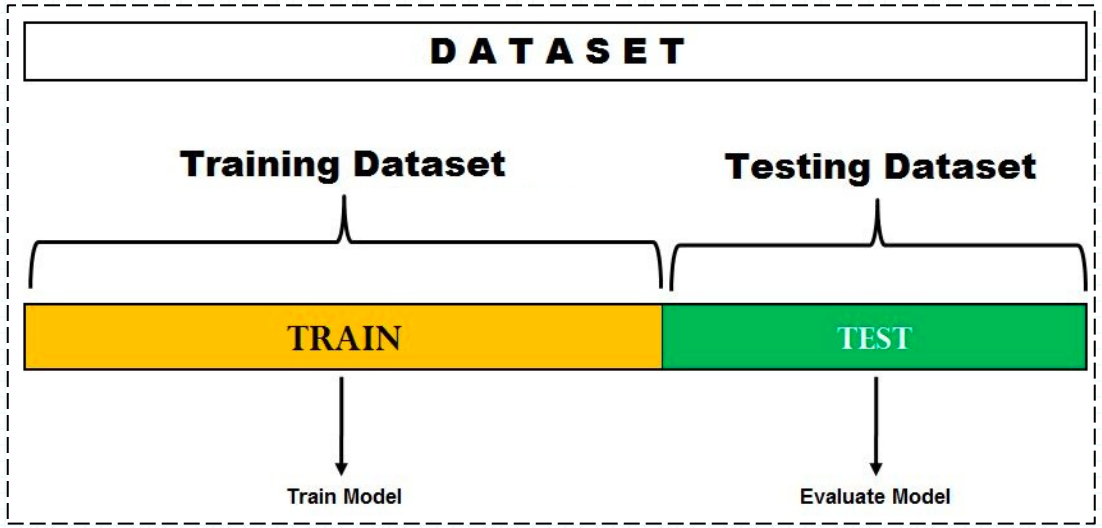
- Hyperparameters are the explicitly specified parameters that control the training process.
- What are the hyperparameters for decision trees?

class `sklearn.tree.`**`DecisionTreeClassifier`**(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0*) [source]

The holdout method

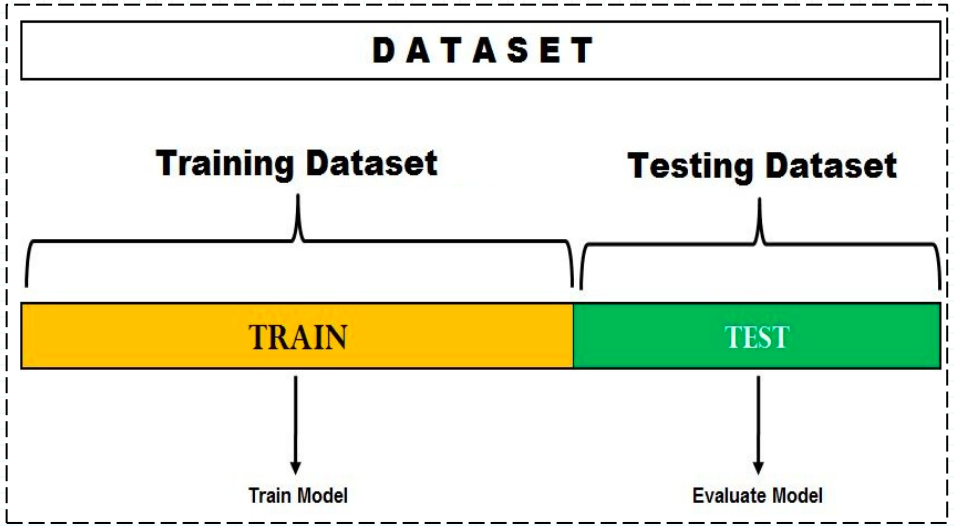


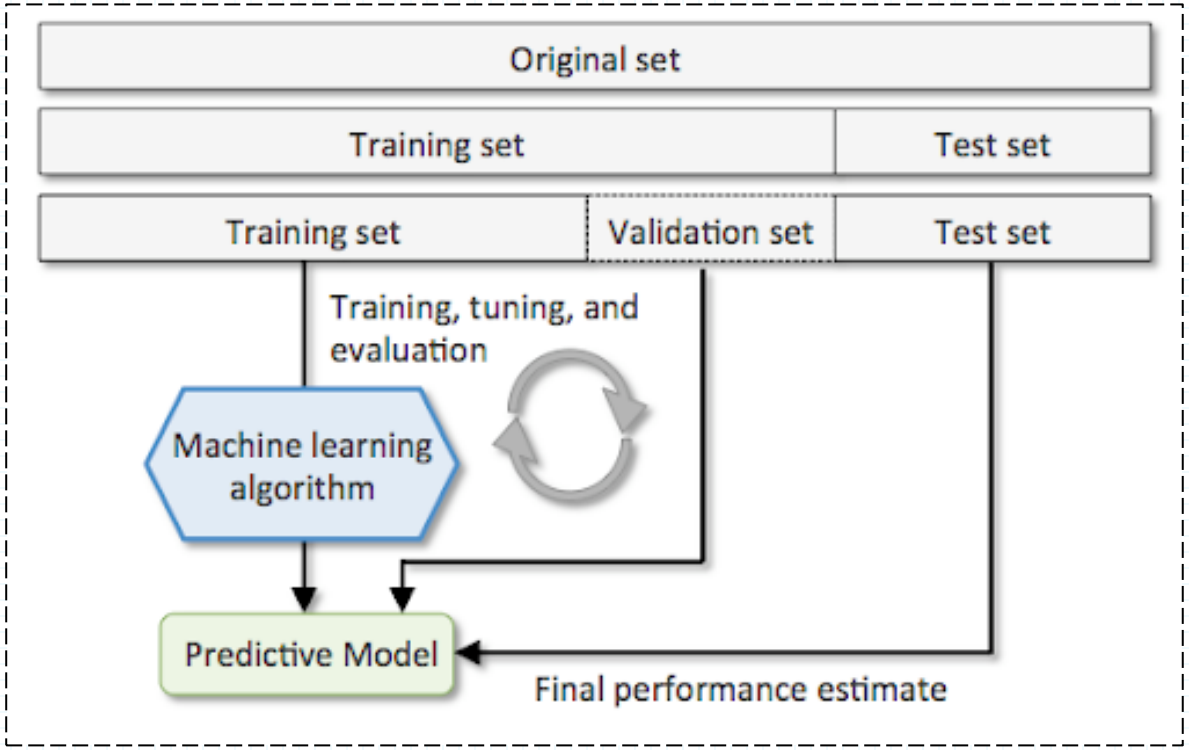
- Split dataset into two groups
  - Training set: used to train the classifier
  - Test set: used to estimate the error rate of the trained classifier.
- Ratio of training and testing sets is at the discretion of analyst;
  - Typically **1:1 or 2:1**, and there is a trade-off between these sizes of these two sets.
  - If the training set is too large, then model may be good enough, but estimation may be less reliable due to small testing set and vice-versa.



Holdout method

But how can we tune hyper-parameters?



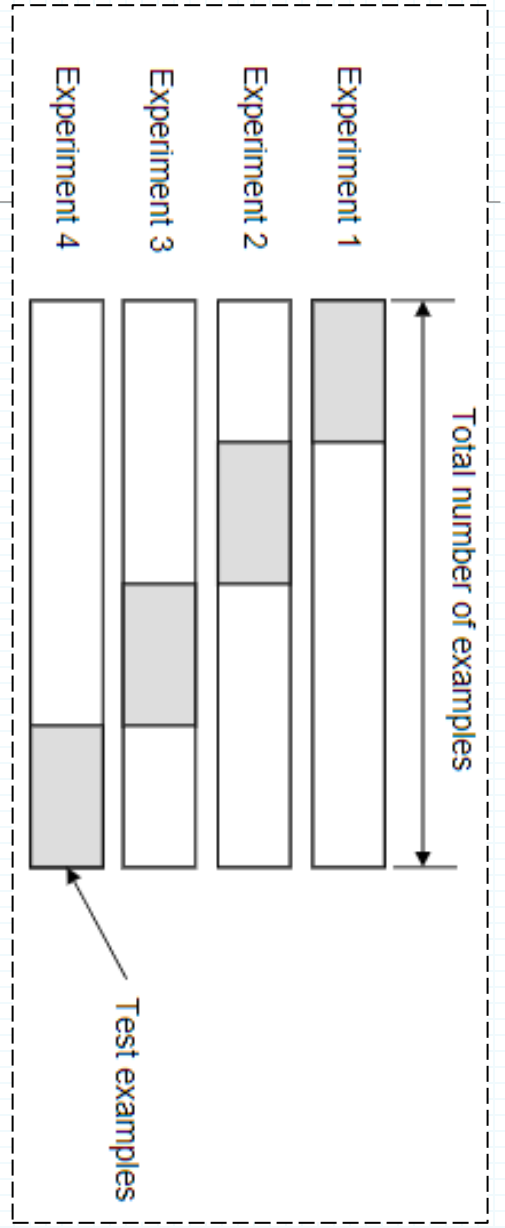


Holdout method - Drawbacks

- The holdout method has **two basic drawbacks**
  - In problems where we have a sparse dataset we may not be able to afford **the "luxury" of setting aside a portion** of the dataset for testing
  - Since it is **a single train-and-test experiment**, the holdout estimate of error rate will be misleading if we happen to get an "unfortunate" split
- The limitations of the holdout can be overcome with a family of re-sampling methods at the expense of higher computational cost
  - K-Fold Cross-Validation
  - Leave-one-out Cross-Validation

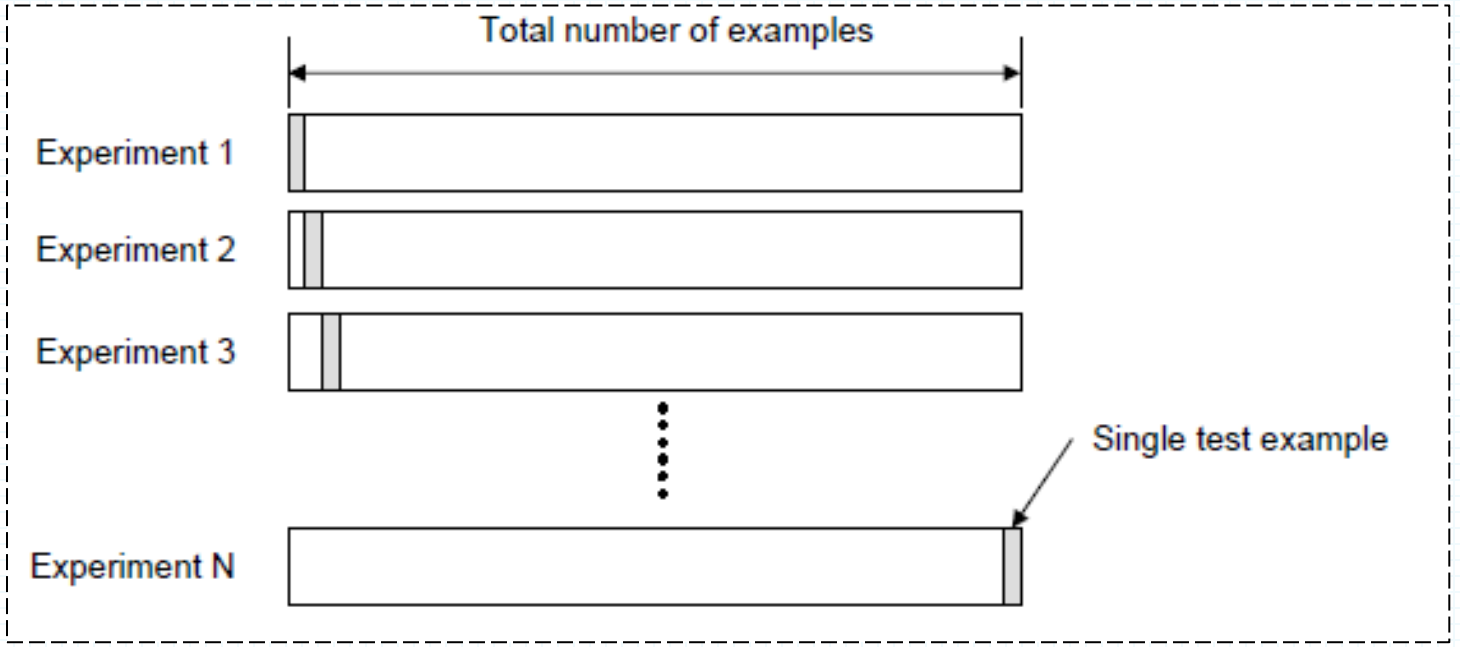
K-Fold Cross-validation

- Create a K-fold partition of the dataset
- For each of K experiments, use K-1 folds for training and a different fold for testing
- The advantage of K-Fold Cross validation is that **all the examples** in the dataset are eventually used for both training and testing
- The true error is estimated as the average error rate on test examples



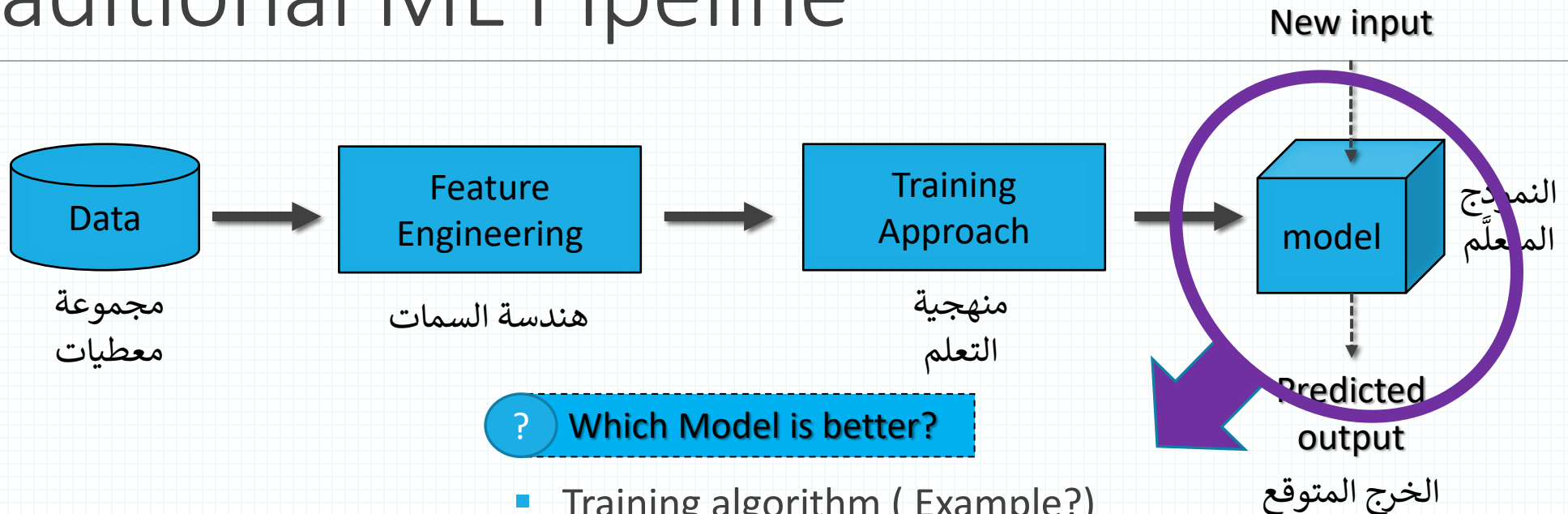
Leave-one-out Cross-Validation

- 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



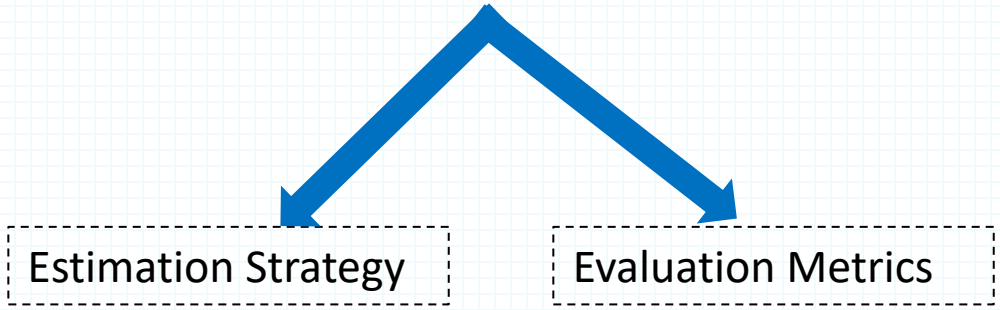
Evaluation Metrics

Traditional ML Pipeline



? Which Model is better?

- Training algorithm (Example?)
- Parameters (Hyperparameters)?
- Handle unseen cases

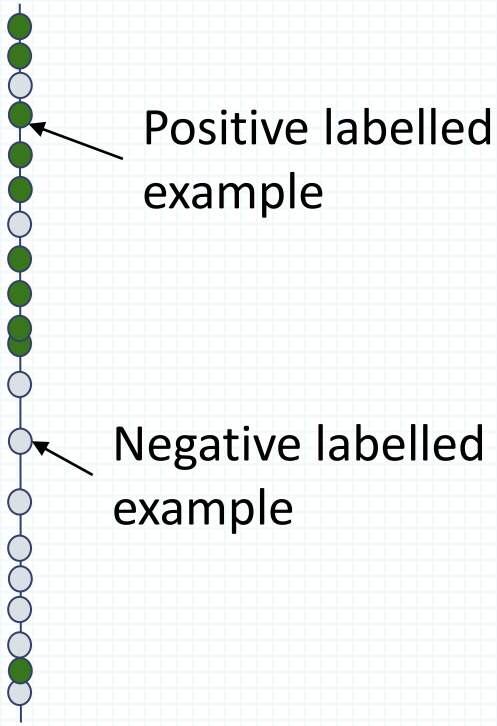


Two types of models

- Models that output a categorical class directly (K Nearest neighbor, Decision tree)
- Models that output a real valued score (SVM, Logistic Regression)
  - Score could be margin (SVM), probability (LR, NN)
  - Need to pick a threshold

Example

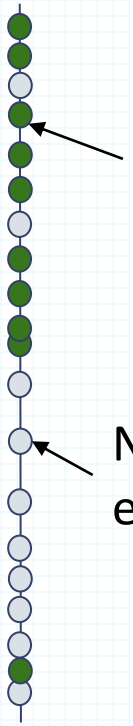
Score = 1



Score = 0

Example

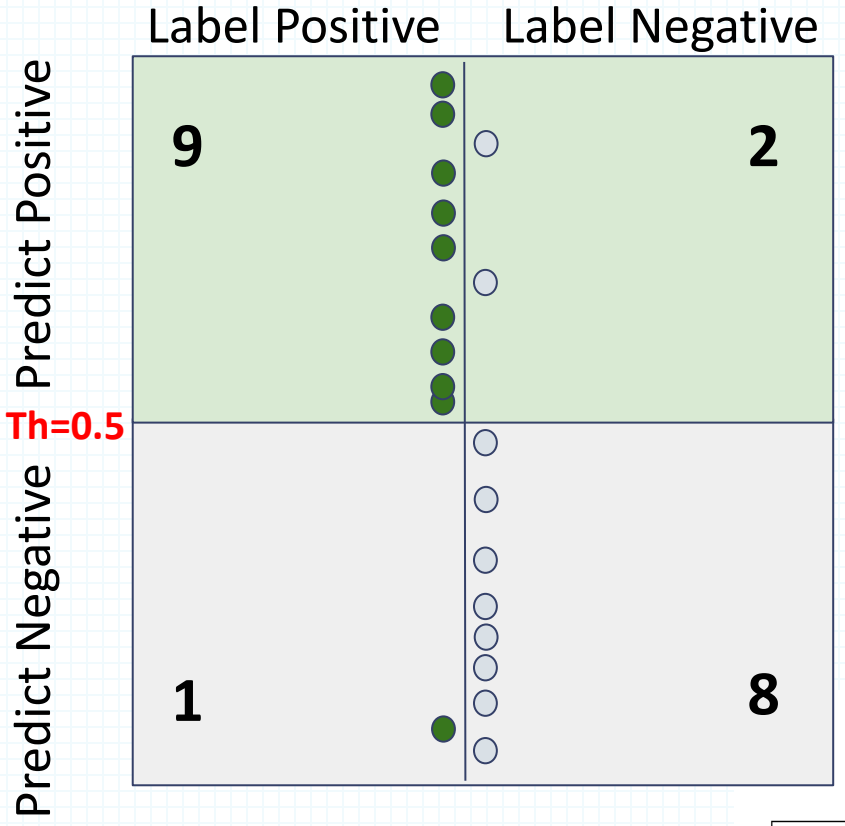
Score = 1



Positive labelled example

Negative labelled example

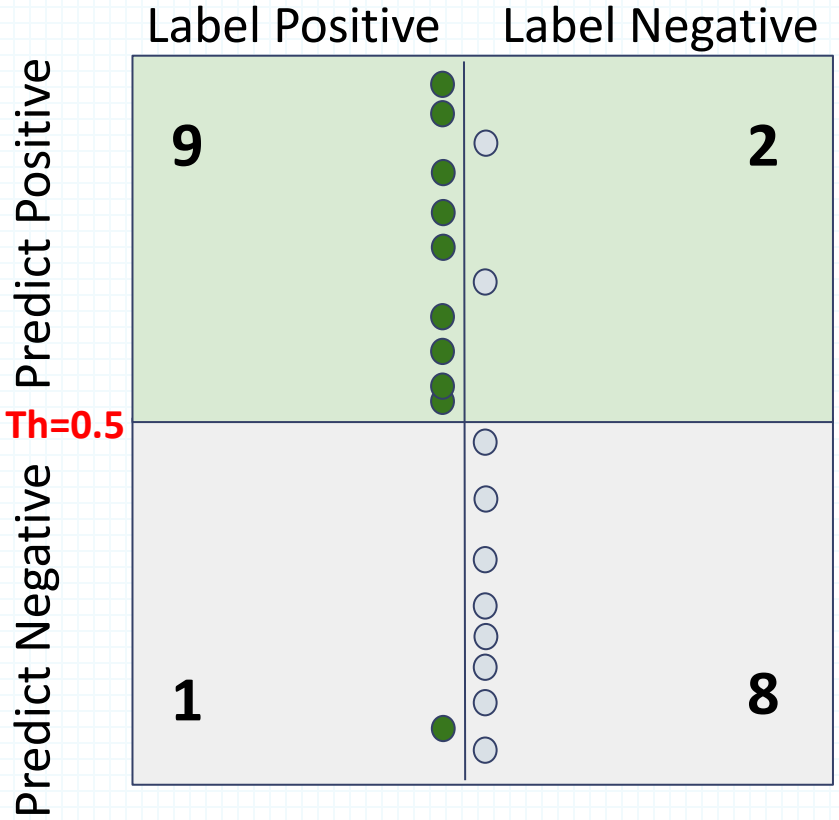
Score = 0



Th	TP	TN	FP	FN
0.5	9	8	2	1

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Evaluation Metrics



TP	TN	FP	FN	Acc
9	8	2	1	0.85

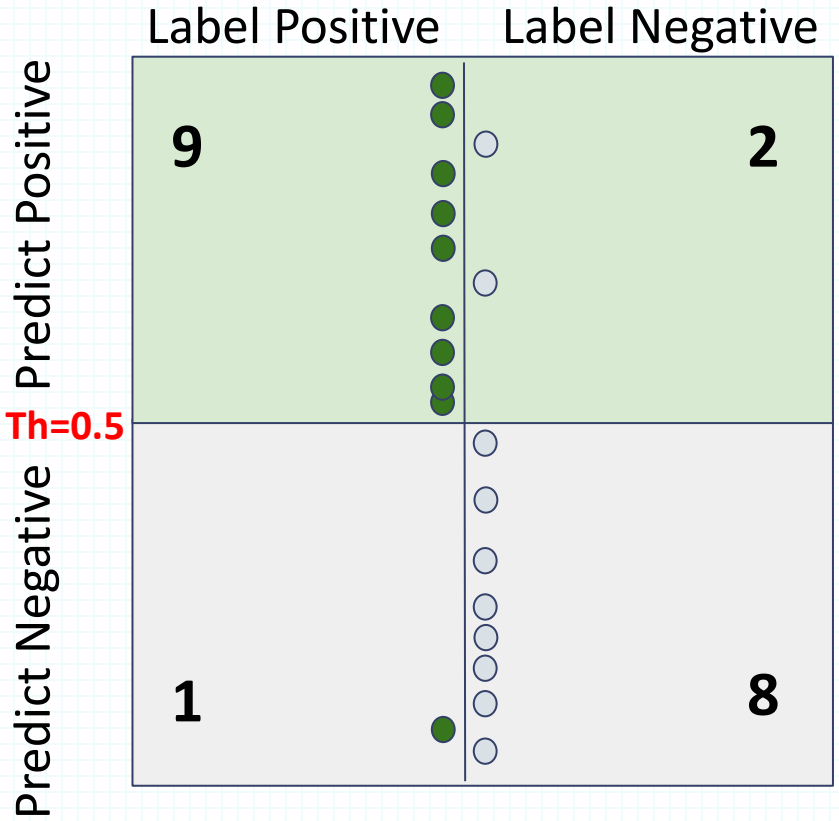
Limitation of Accuracy

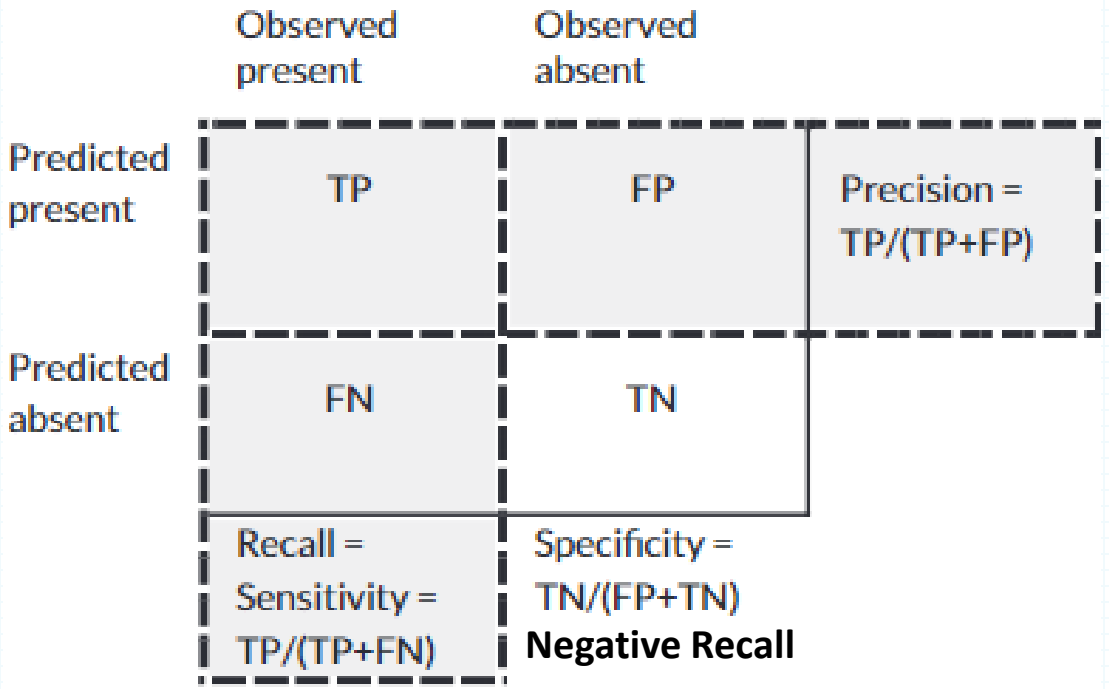
- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
- Accuracy is misleading because model does not detect any class 1 example

Evaluation Metrics

TP	TN	FP	FN	Acc	P	R	F1
9	8	2	1	0.85	0.81	0.90	0.857





F1-score

- It is usually better to compare models by means of one number only.
- The F1−score can be used to combine precision and recall

	Precision(P)	Recall (R)	Average	F_1 Score
Algorithm 1	0.5	0.4	0.45	0.444
Algorithm 2	0.7	0.1	0.4	0.175
Algorithm 3	0.02	1.0	0.51	0.0392

The best is Algorithm 1

Algorithm 3 predict always 1

Average says not correctly that Algorithm 3 is the best

$$\text{Average} = \frac{P + R}{2} \qquad F_1\text{score} = 2\frac{PR}{P + R}$$

- $P = 0$ or $R = 0 \Rightarrow F_1\text{score} = 0$
- $P = 1$ and $R = 1 \Rightarrow F_1\text{score} = 1$

Multiclass Classifier

- Having m classes, confusion matrix is a table of size m×m , where, element at (i, j) indicates the number of instances of class i but classified as class j.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	52	10	7	0	0	1
C_2	15	50	6	2	1	2
C_3	5	6	6	0	0	0
C_4	0	2	0	10	0	1
C_5	0	1	0	0	7	1
C_6	1	3	0	1	0	24

- What is the accuracy?

Multiclass Classifier

- Precision, Recall, and F1-score are calculated for each class.
  - A large confusion matrix of m*m can be considered into 2*2 matrix.

Class	C_1	C_2	C_3	C_4	C_5	C_6
C_1	52	10	7	0	0	1
C_2	15	50	6	2	1	2
C_3	5	6	6	0	0	0
C_4	0	2	0	10	0	1
C_5	0	1	0	0	7	1
C_6	1	3	0	1	0	24

C1	+	-
+	52	18
-	21	123

- Finally, we can merge overall scores (ex: using weighted average)
  - What is the overall F1-score in the previous example?

More Evaluation Metrics

- **Classification:** Precision, Recall, AUC
- **Ranking / Retrieval:** Precision@K, Recall@K
- **Segmentation:** IoU
- **Regression:** MAE, RMSE
- **Text Generation:** BLEU, ROUGE