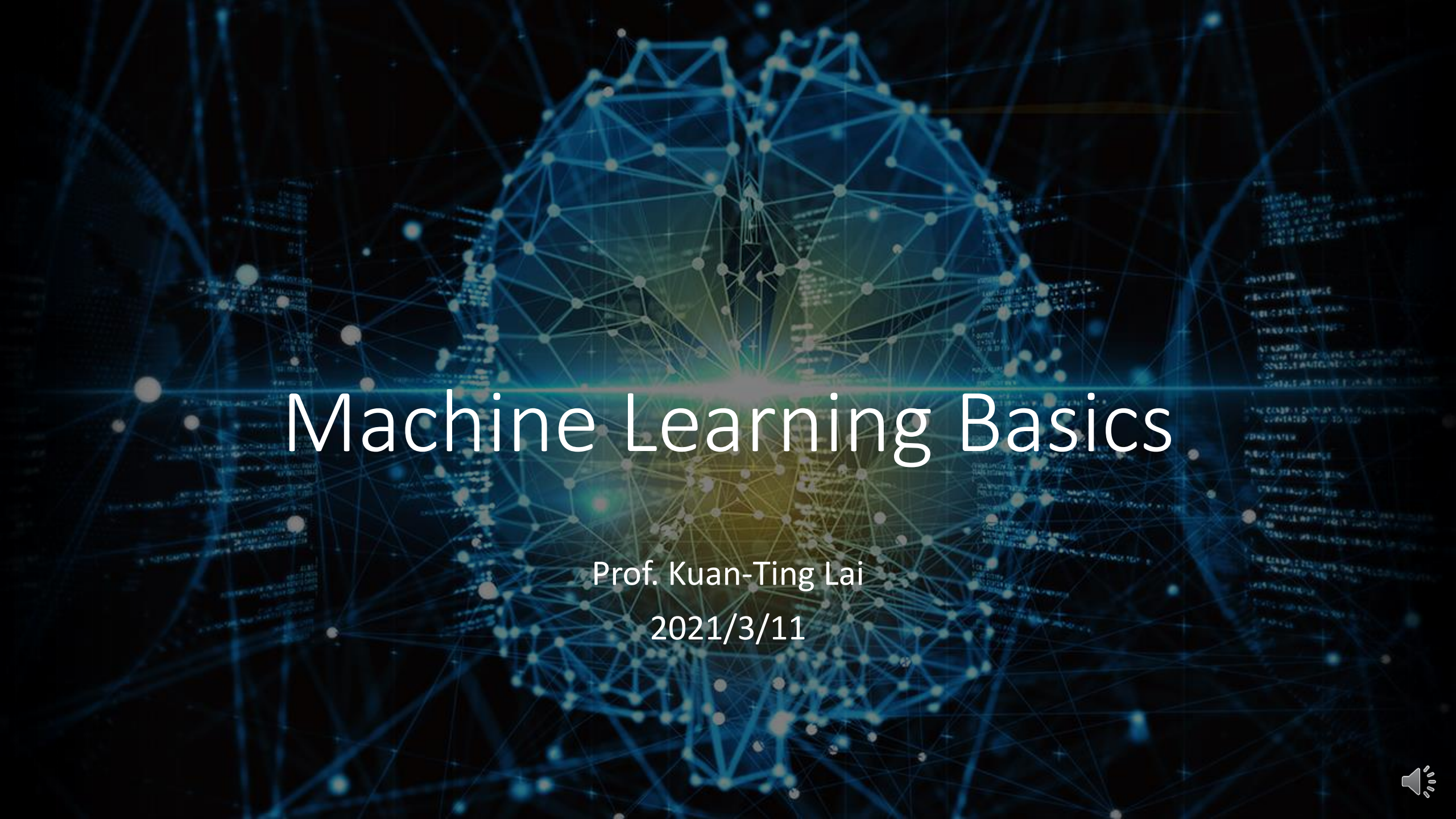
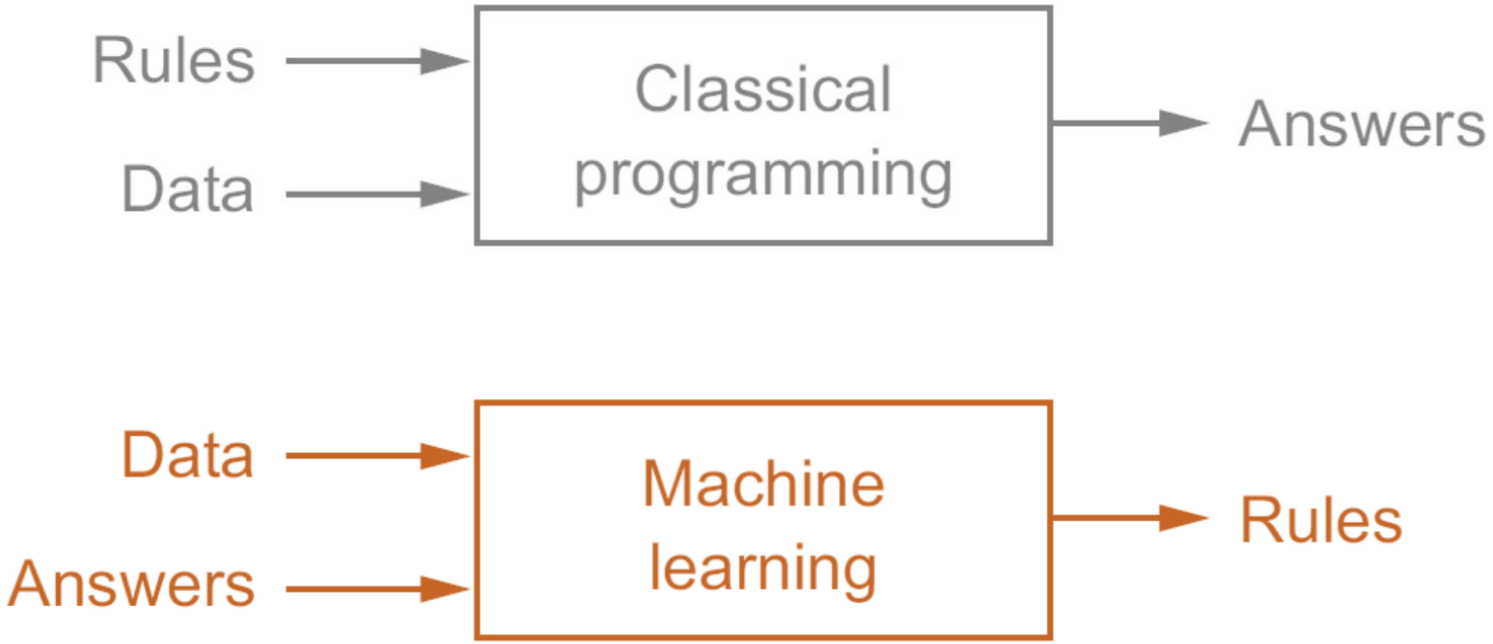


Machine Learning Basics

Prof. Kuan-Ting Lai

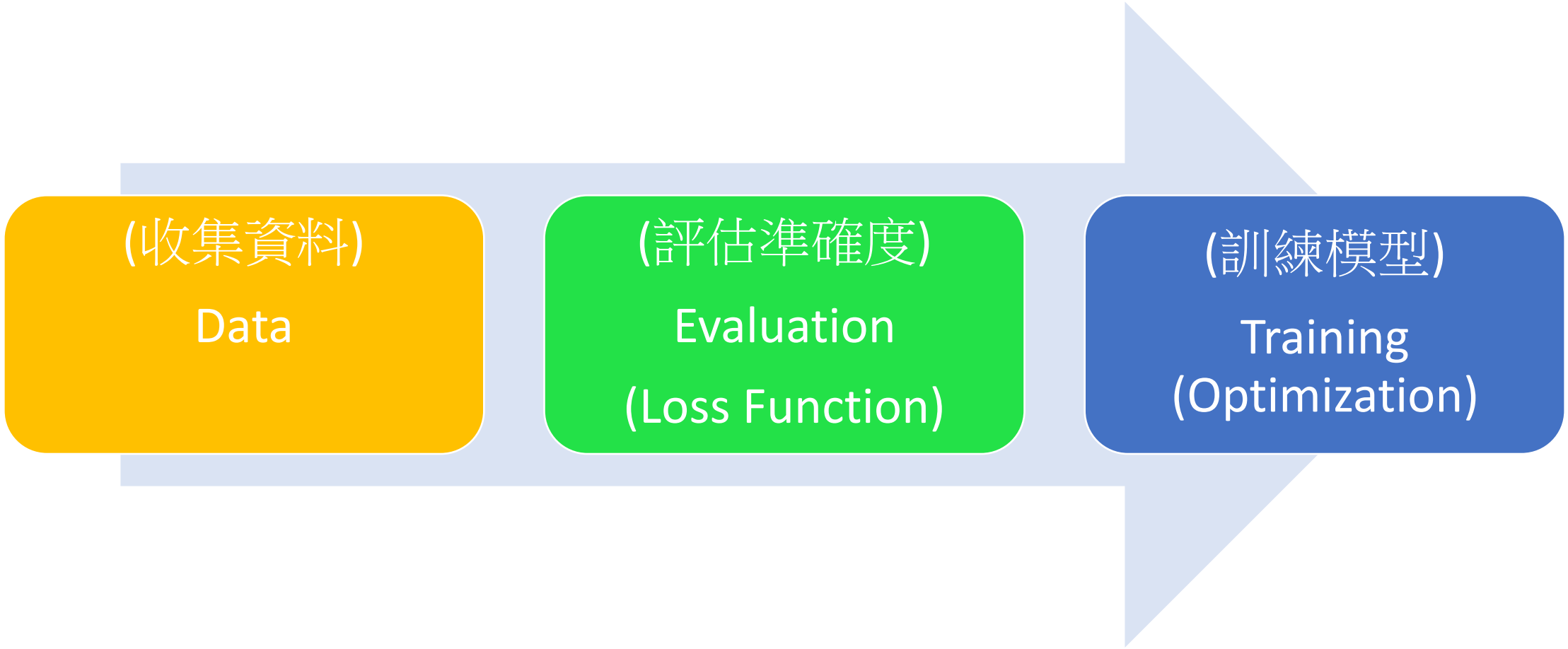
2021/3/11

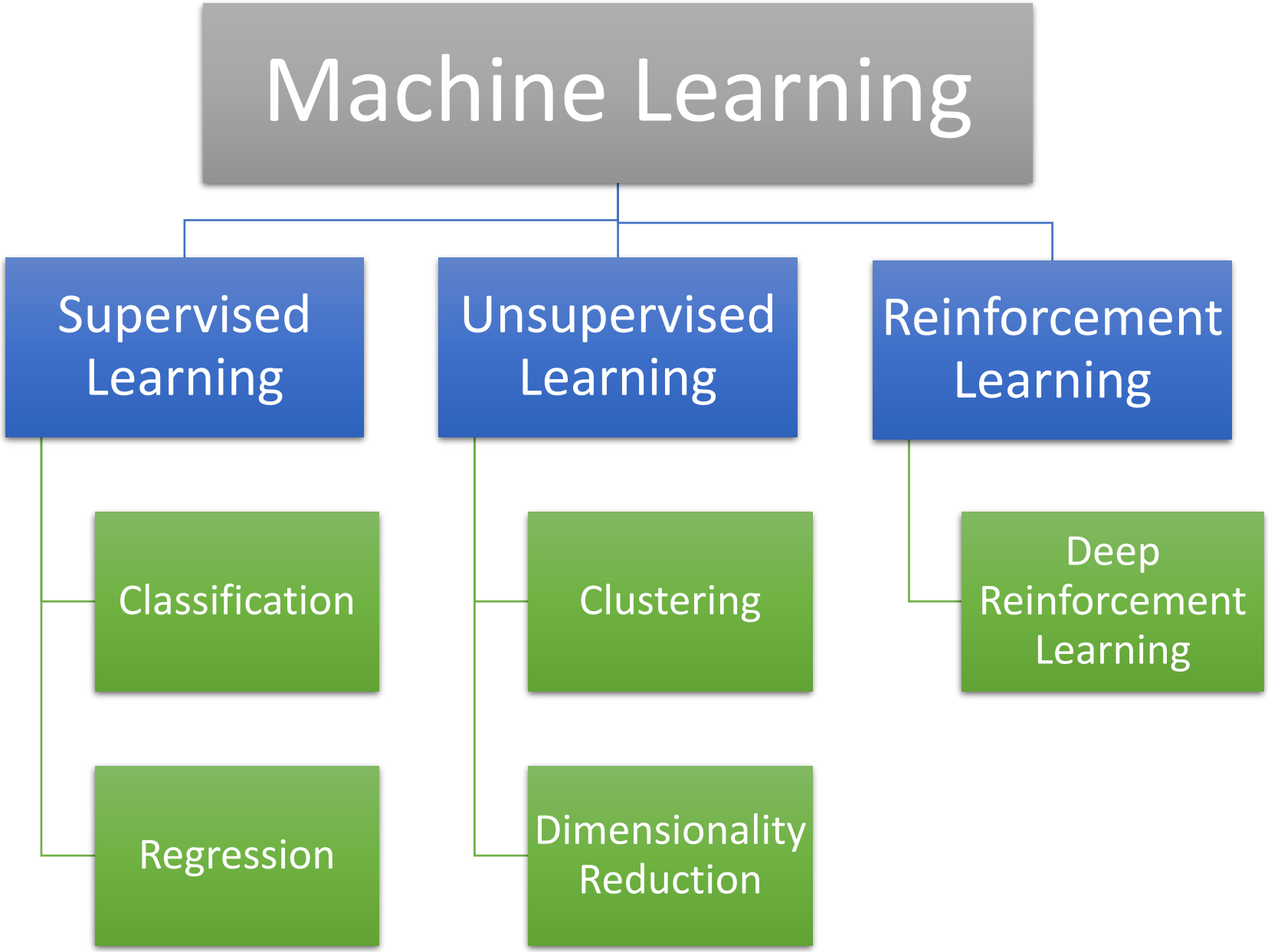
Machine Learning



Francois Chollet, “Deep Learning with Python,” Manning, 2017

Machine Learning Flow

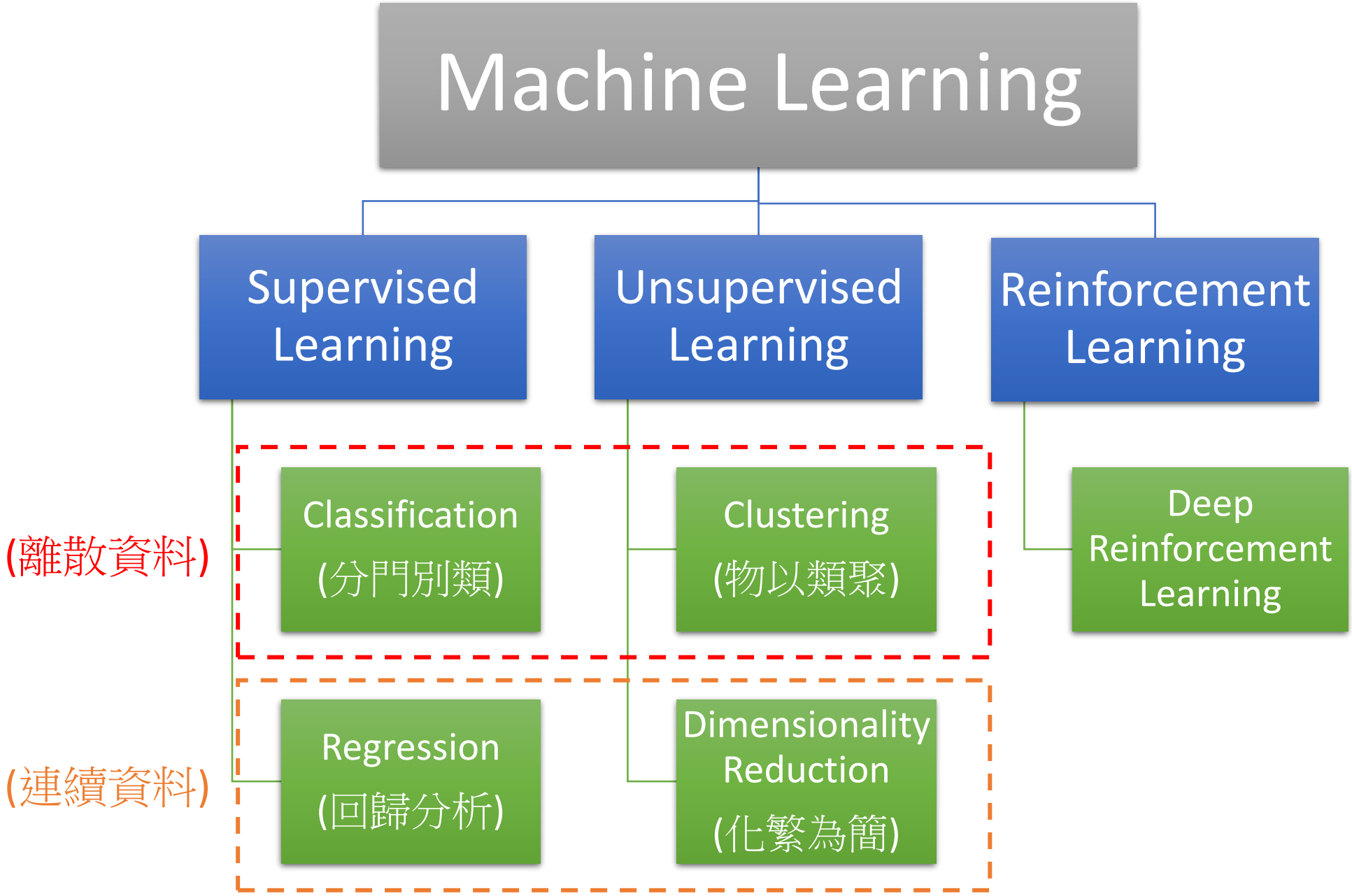


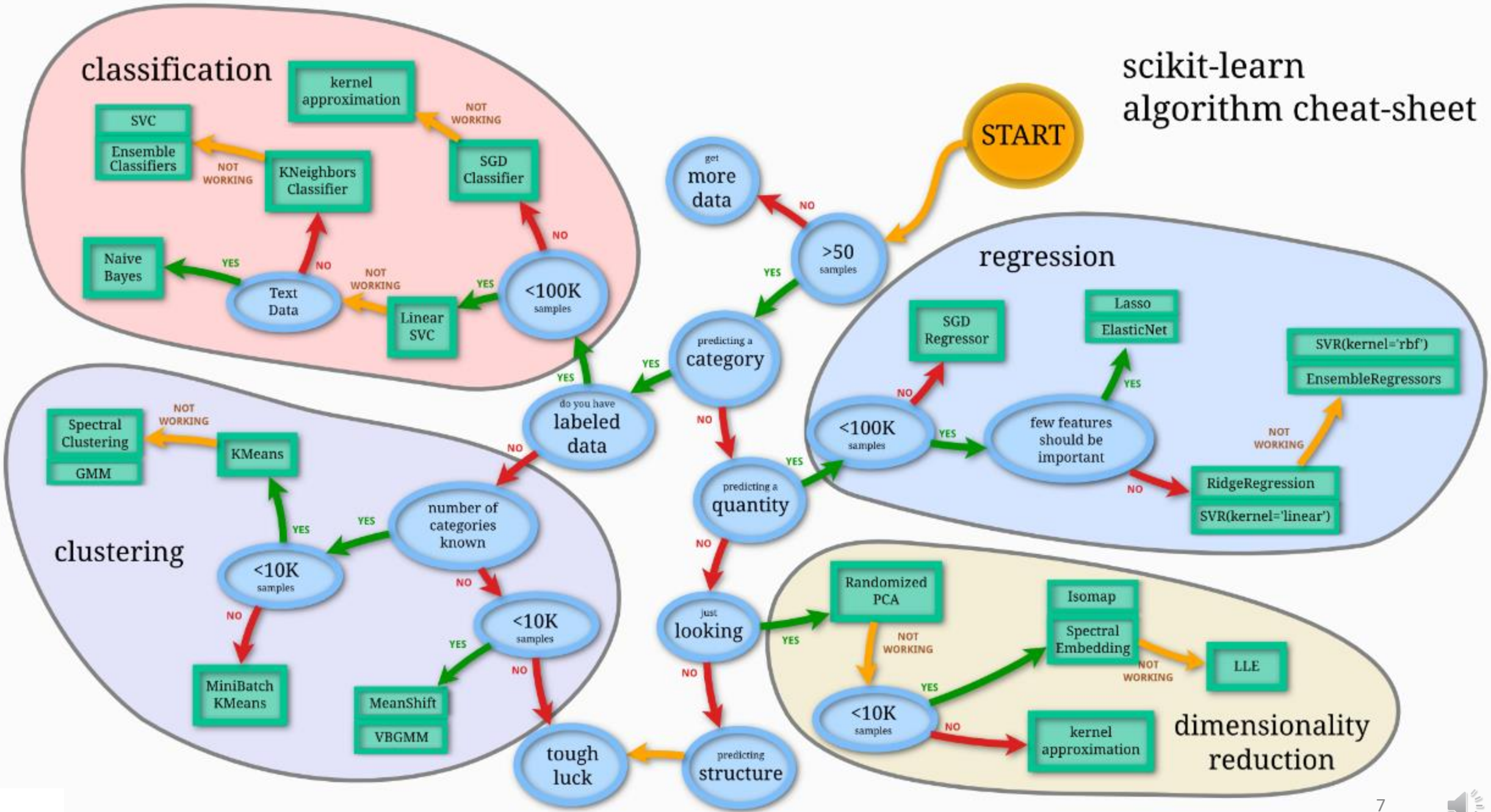

Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Classification
Regression
Clustering
Dimensionality Reduction
Deep Reinforcement Learning

Machine Learning

Has a teacher to label data!

- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
  - Clustering
  - Dimensionality Reduction
- Reinforcement Learning
  - Deep Reinforcement Learning


Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Classification
(分門別類)
Clustering
(物以類聚)
Deep Reinforcement Learning
Regression
(回歸分析)
Dimensionality Reduction
(化繁為簡)
(離散資料)
(連續資料)

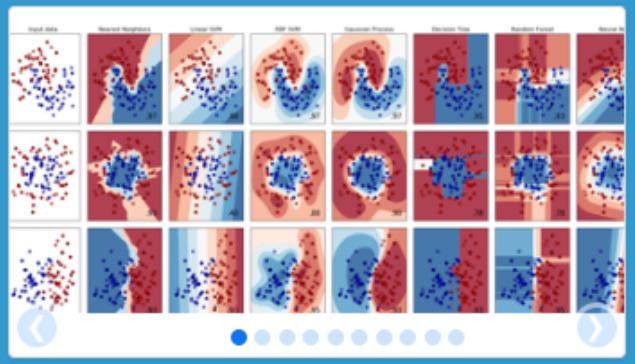

scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
YES
do you have labeled data
YES
<100K samples
YES
Linear SVC
NOT WORKING
Text Data
YES
Naive Bayes
NO
KNeighbors Classifier
NOT WORKING
Ensemble Classifiers
SVC
NOT WORKING
kernel approximation
NO
SGD Classifier
classification
NO
number of categories known
YES
<10K samples
YES
KMeans
NOT WORKING
Spectral Clustering
GMM
NO
MiniBatch KMeans
NO
<10K samples
YES
MeanShift
VBGMM
NO
tough luck
clustering
NO
predicting structure
NO
just looking
NO
predicting a quantity
YES
<100K samples
NO
SGD Regressor
YES
few features should be important
YES
Lasso
ElasticNet
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
regression
YES
Randomized PCA
NOT WORKING
<10K samples
YES
Isomap
Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
dimensionality reduction

scikit-learn.org

scikit learn

Home Installation Documentation Examples

Google Custom Search



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

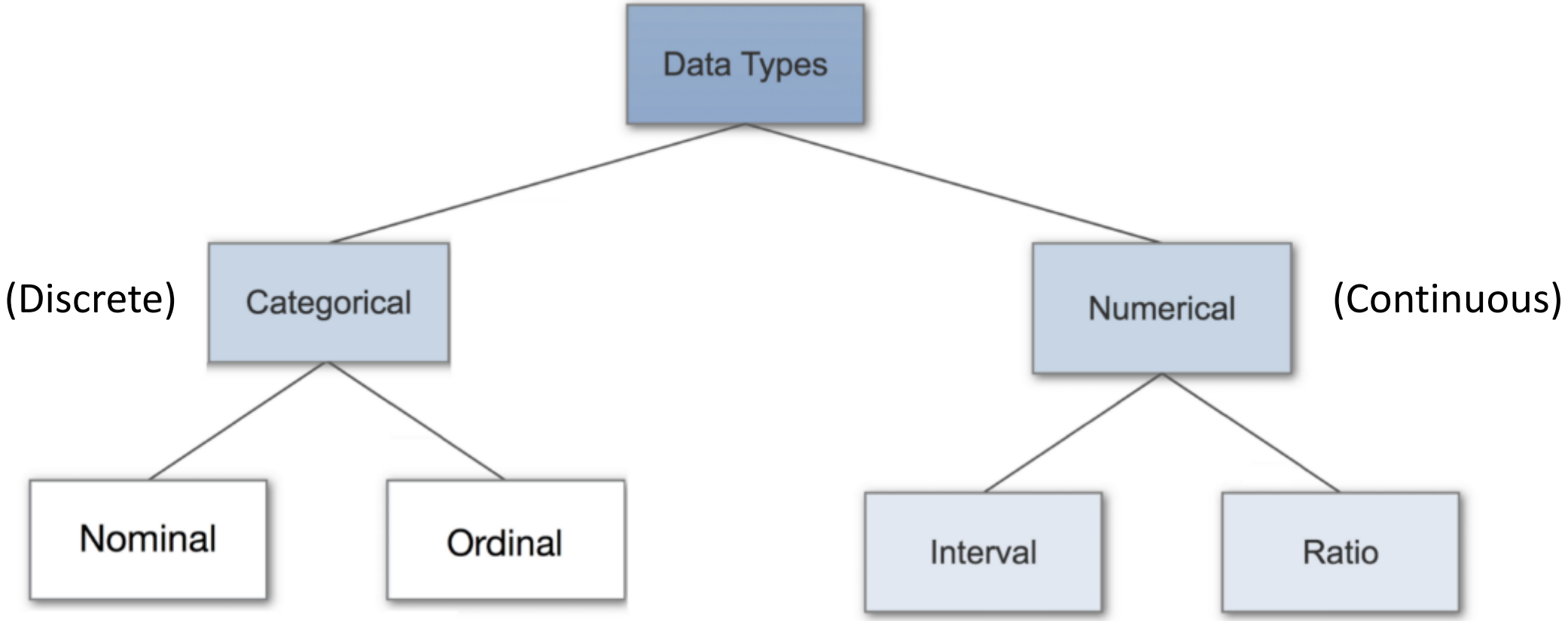
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.

— Examples

Types of Data

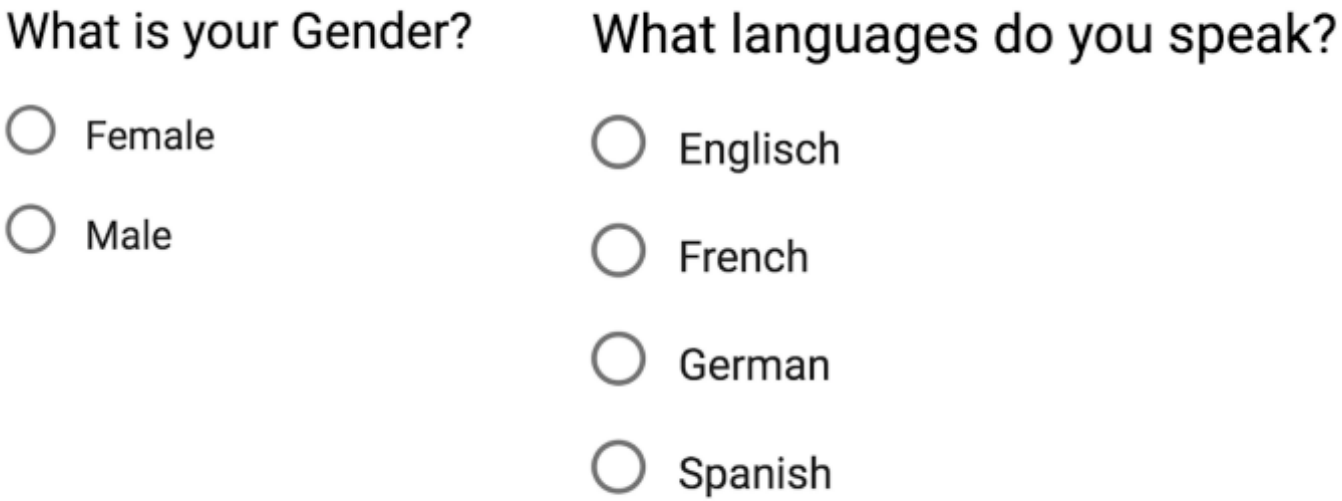
Data Types (Measurement Scales)



https://towardsdatascience.com/data-types-in-statistics-347e152e8bee

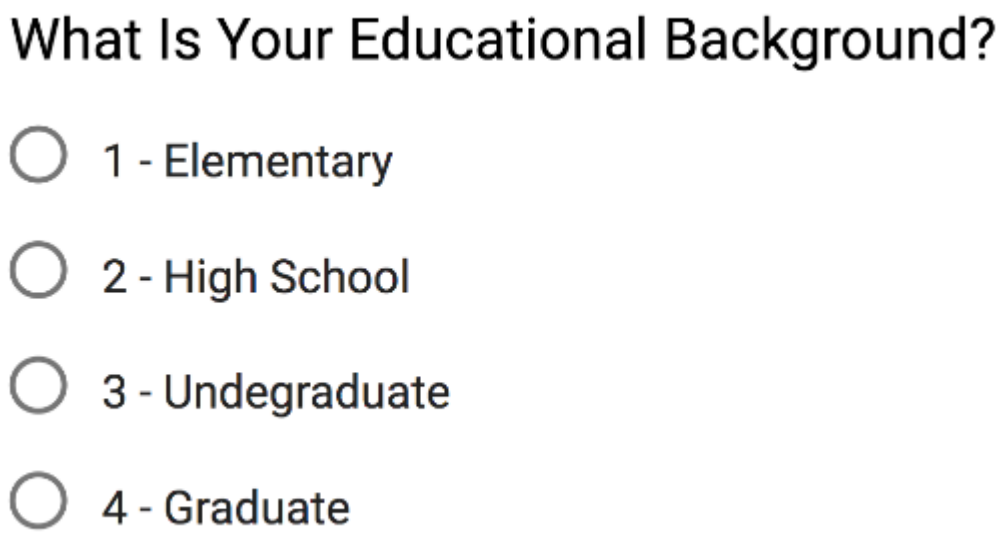
Nominal Data (Labels)

- Nominal data are labeling variables without any quantitative value
- Encoded by one-hot encoding for machine learning
- Examples:



Ordinal Data

- Ordinal values represent discrete and ordered units
- The order is meaningful and important



Interval Data

- Interval values represent **ordered units that have the same difference**
- Problem of Interval: **Don’t have a true zero**
- Example: Temperature Celsius (°C) vs. Fahrenheit (°F)

Temperature?

- ○ - 10
- ○ -5
- ○ 0
- ○ + 5
- ○ + 10
- ○ + 15

Ratio Data

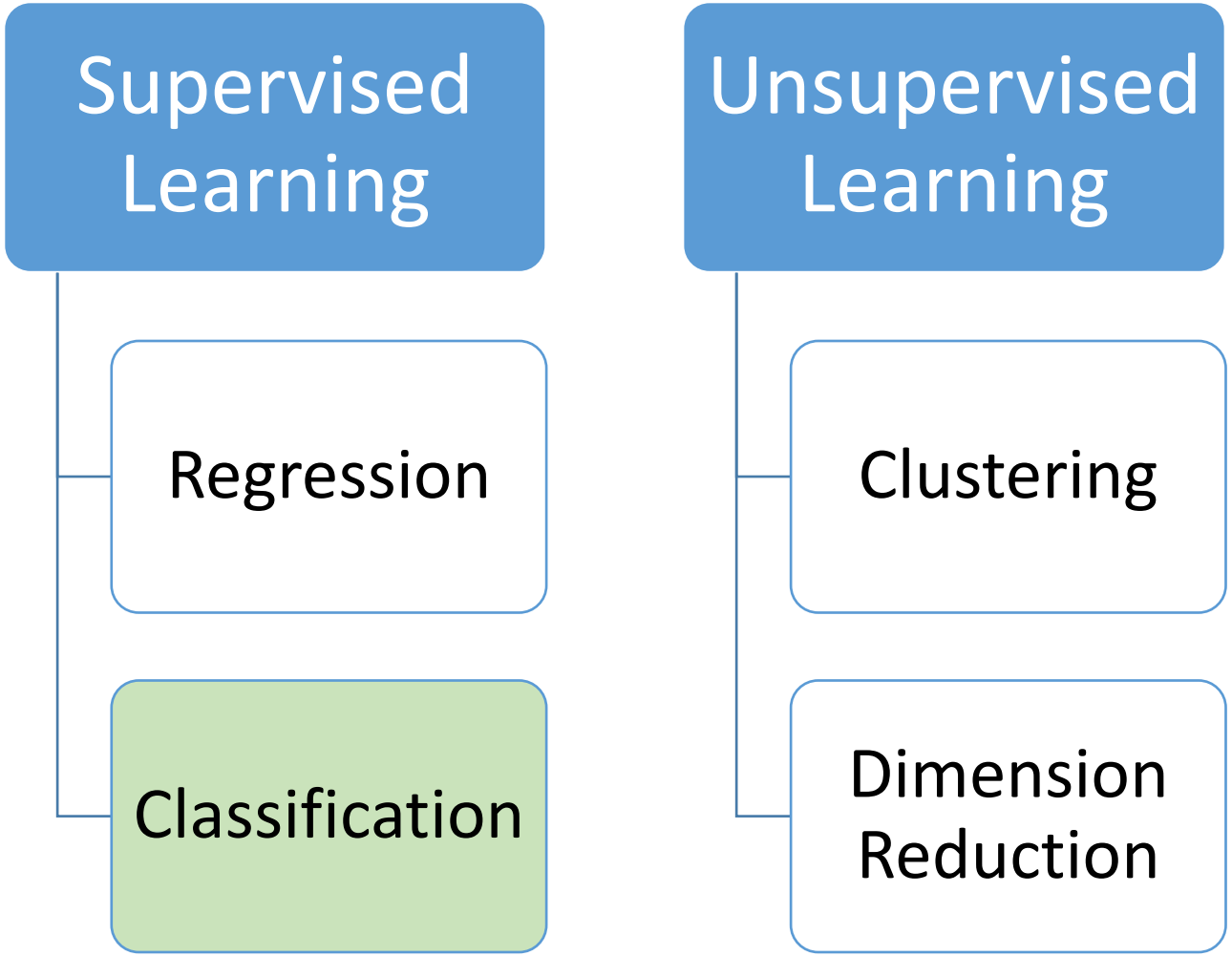
- Same as interval data but have absolute zero
- Can be applied to both descriptive and inferential statistics
- Example: weight & height

Machine Learning vs. Statistics

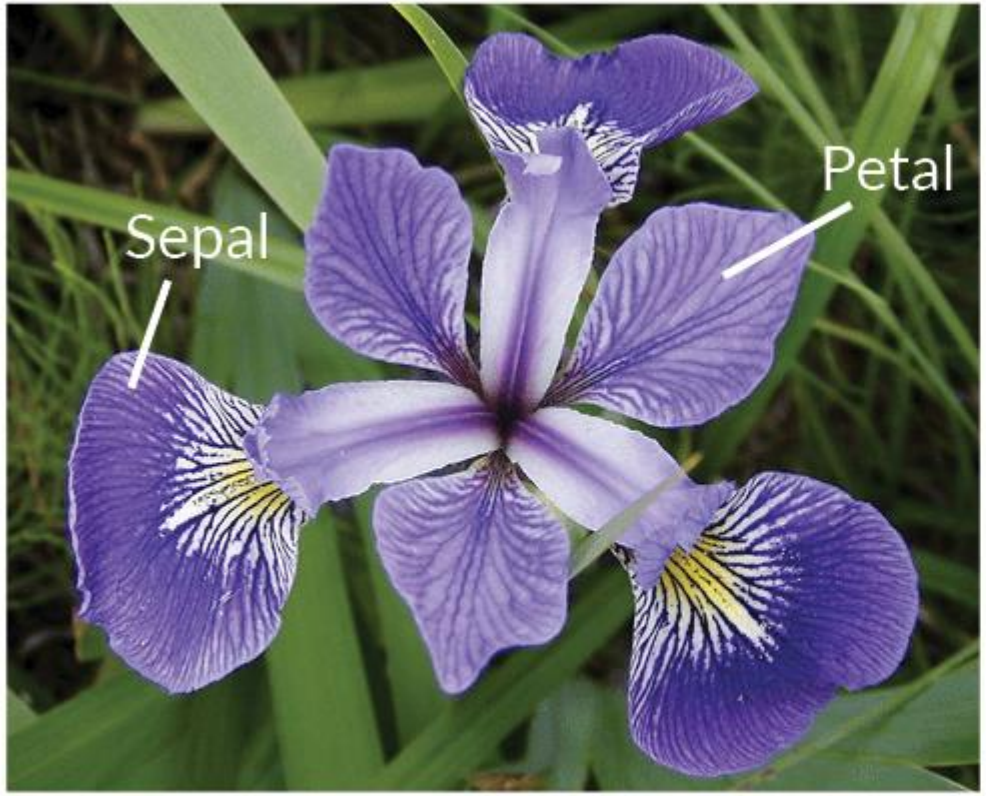
- https://www.r-bloggers.com/whats-the-difference-between-machine-learning-statistics-and-data-mining/

Machine learning	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = $1,000,000	large grant = $50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

Supervised and Unsupervised Learning



Iris Flower Classification (鳶尾花分類)



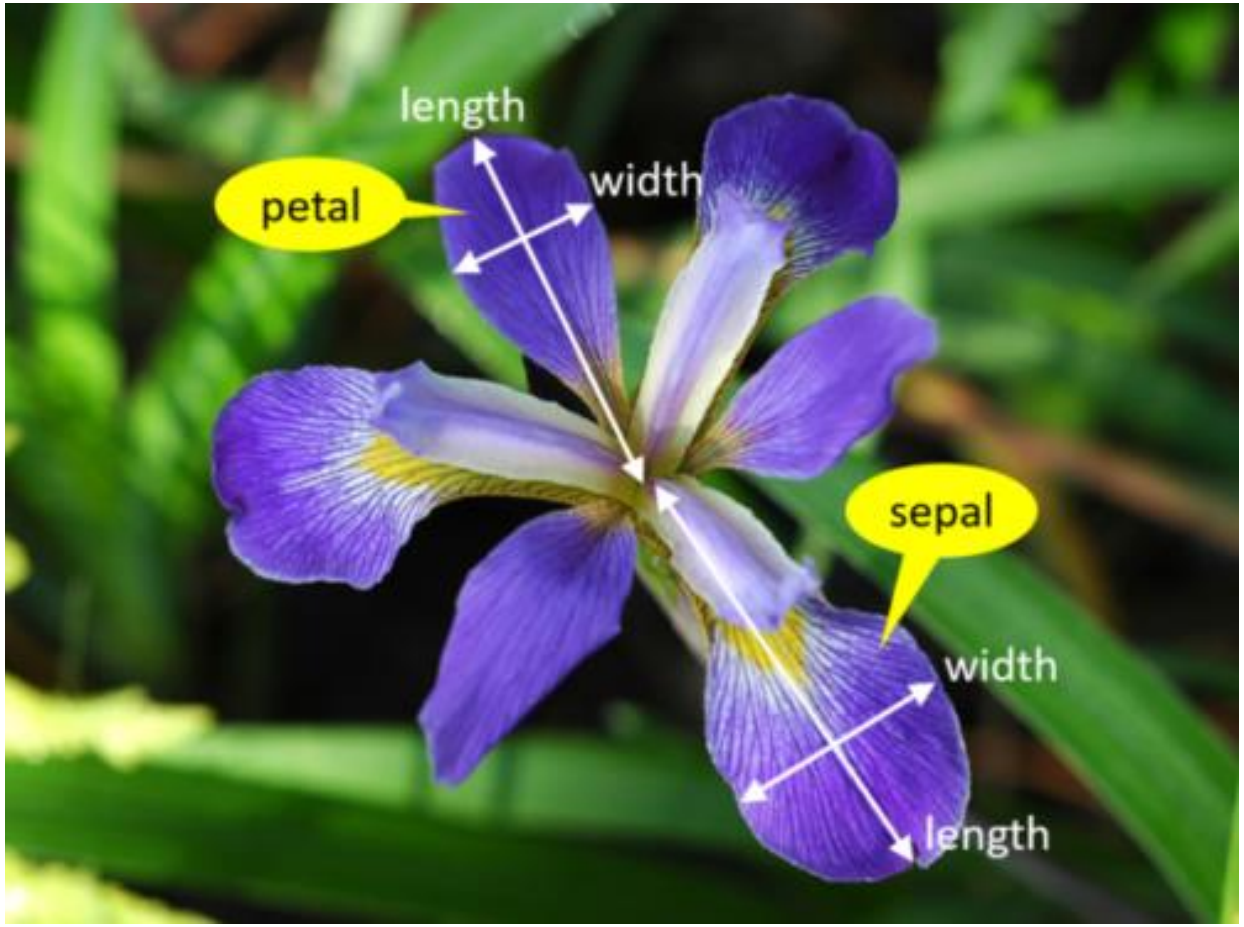
Iris Versicolor

Iris Setosa

Iris Virginica

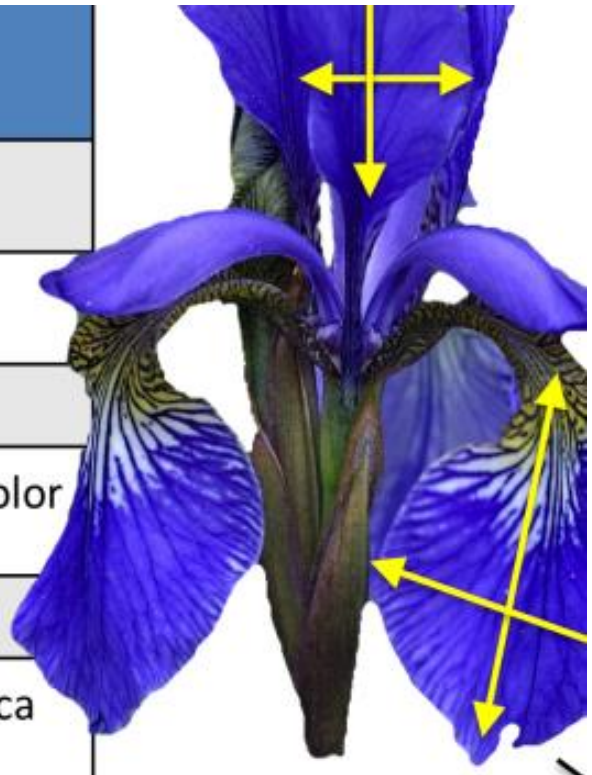
Extracting Features of Iris (抽取特徵值)

- Width and Length of Petal (花瓣) and Sepal (花萼)



Iris Flower Dataset

	Sepal length	Sepal width	Petal length	Petal width	Class label
	5.1	3.5	1.4	0.2	Setosa
	4.9	3.0	1.4	0.2	Setosa
			...		
)	6.4	3.5	4.5	1.2	Versicolor
			...		
50	5.9	3.0	5.0	1.8	Virginica



eatures

ttributes, measurements, dimensions)

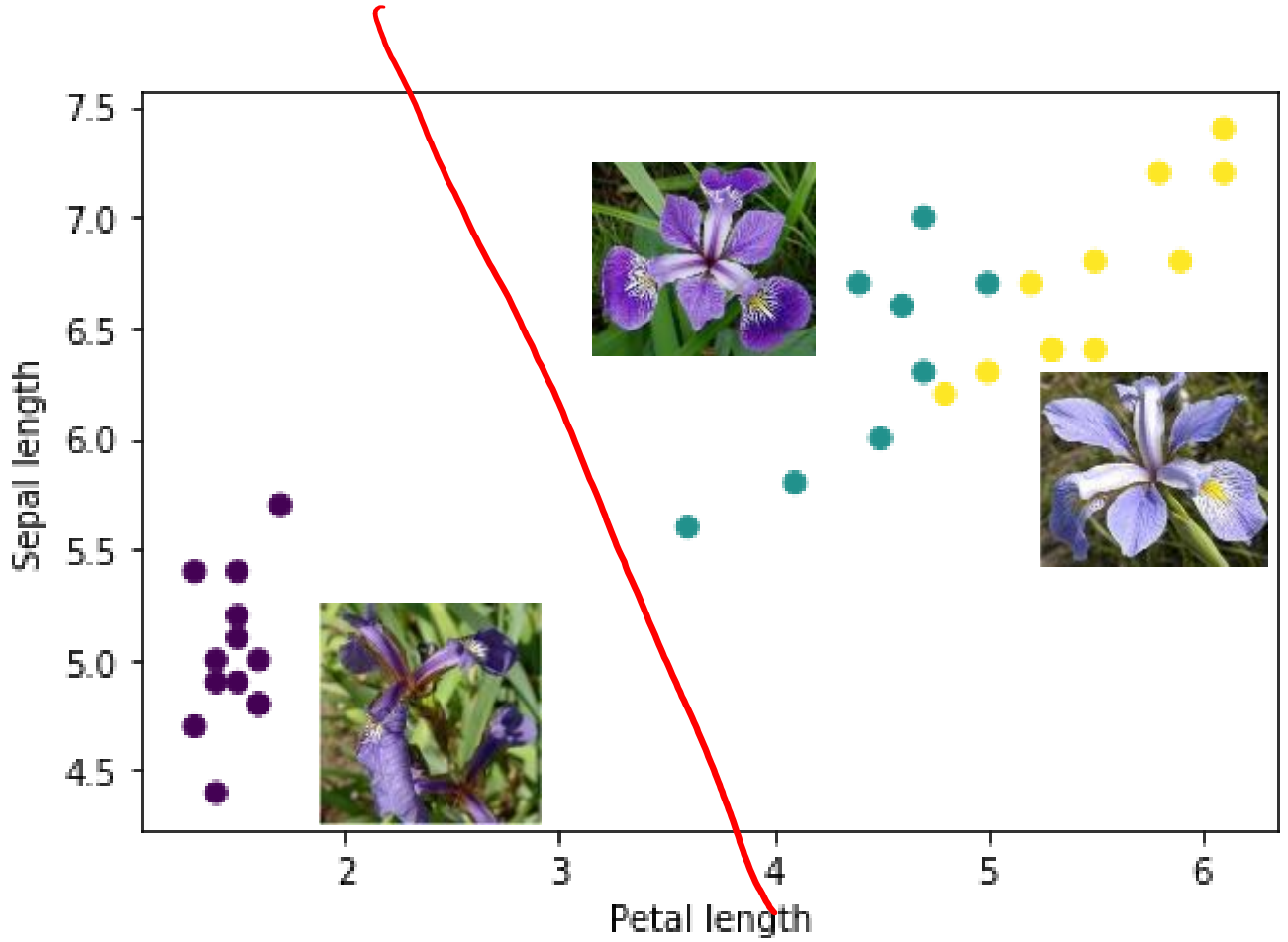
Class labels

(targets)

[Jebaseelan Ravi @ Medium]

Classify Iris Species via Petals and Sepals

- Iris versicolor and virginica are not linearly separable



https://www.tensorflow.org/tutorials/customization/custom_training_walkthrough

Linear Classifier

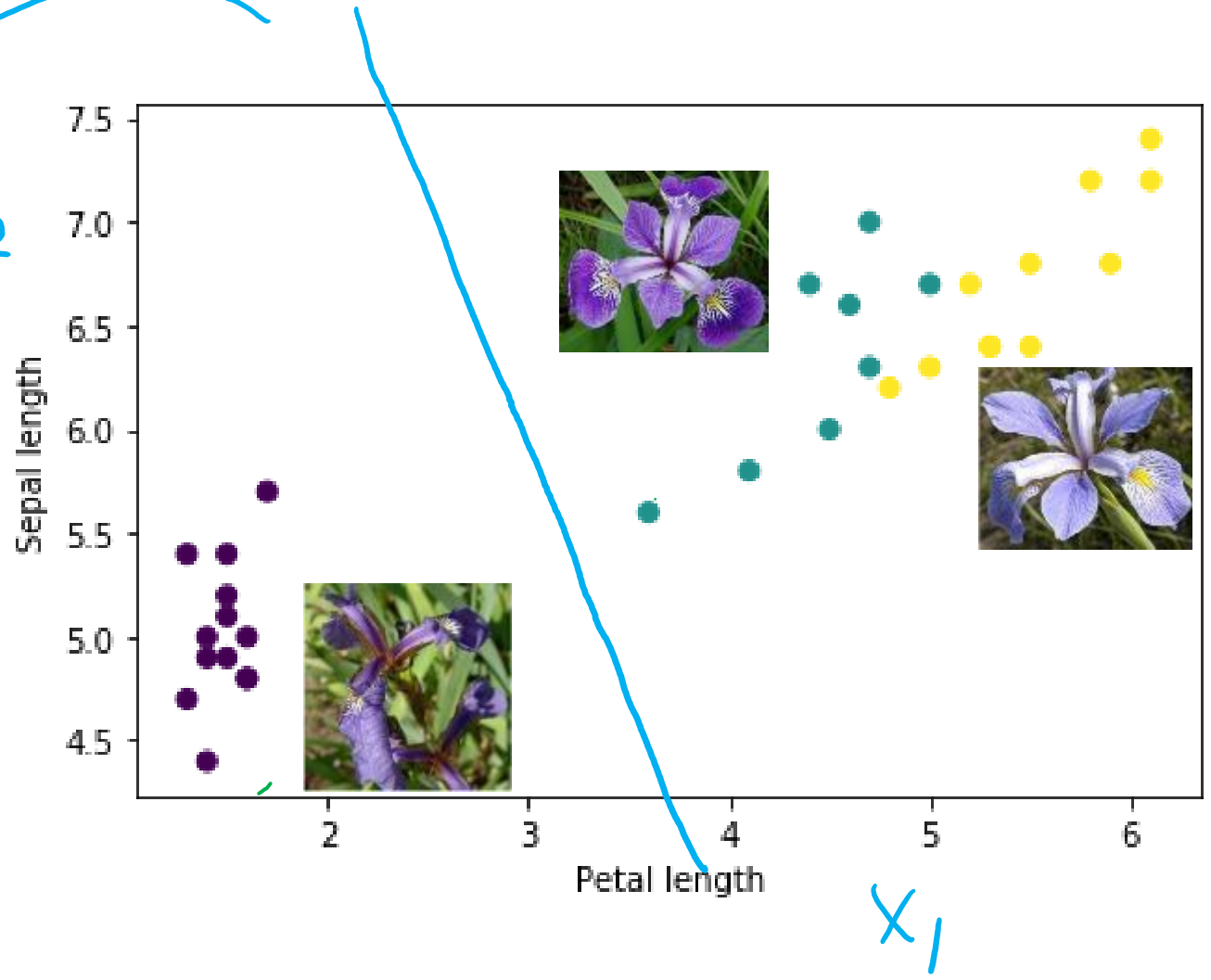
$$w_1 x_1 + w_2 x_2 - b = 0$$

$$[w_1 \; w_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - b = 0$$

$$w^T x - b = 0$$

$$w^T x - b \leq 0 \quad \text{Setosa}$$

$$w^T x - b > 0 \quad \text{Versicolor, Virginica}$$



Evaluation (Loss Function)

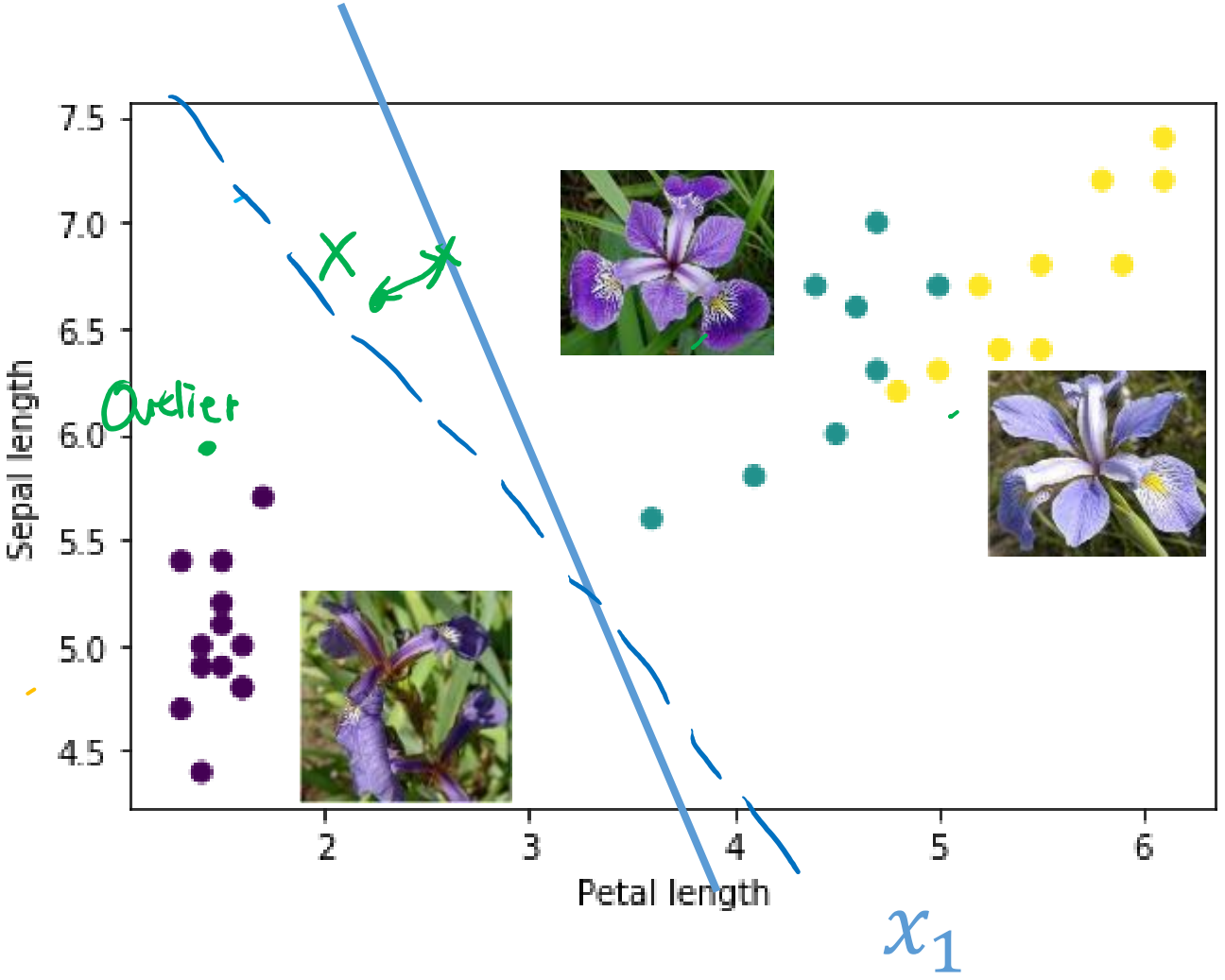
$$y' = w^T x - b < 0$$

$$\Delta = y - y'$$

Learning Rule (Perceptron)

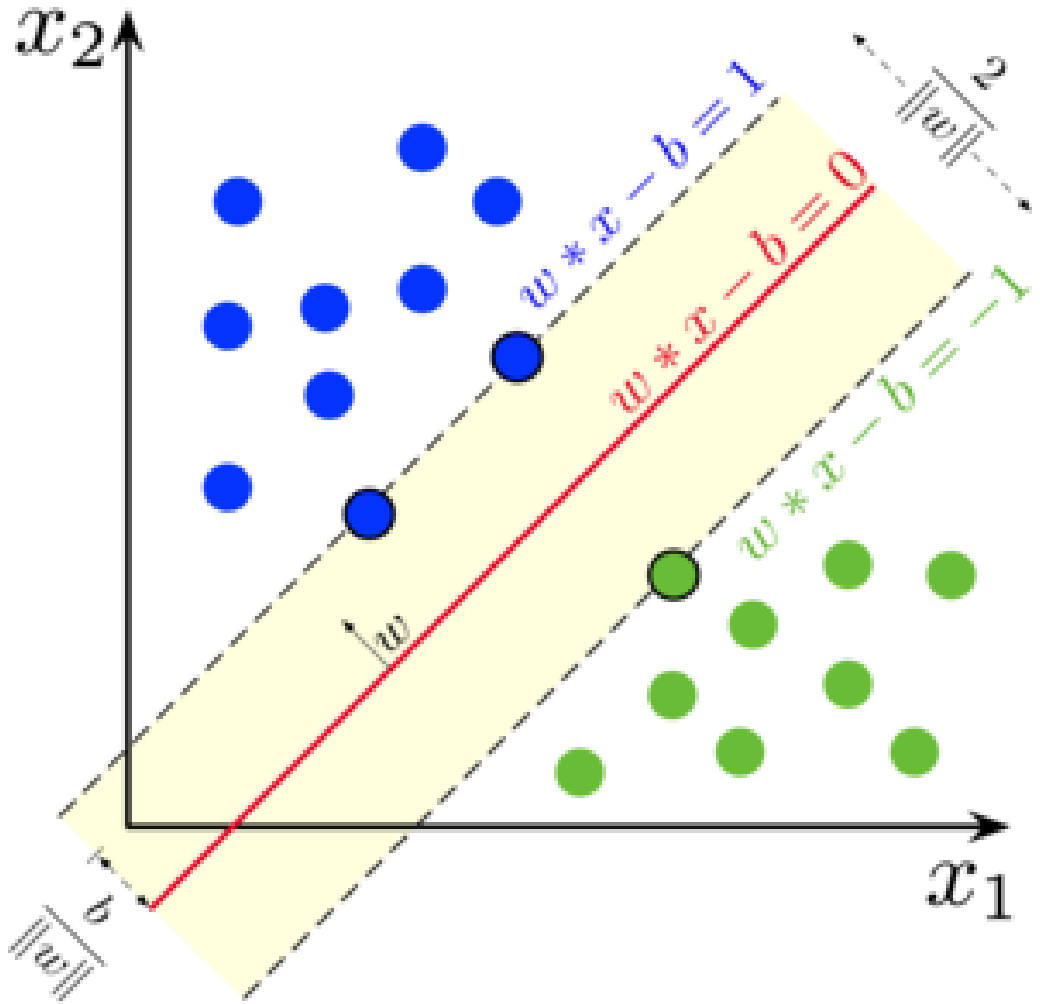
$$w \leftarrow w + \alpha(y - y')x$$

↑ Learning Rate



Support Vector Machine (SVM)

- Choose the hyperplanes that have the largest separation (margin)



Loss Function of SVM

- Calculate prediction errors

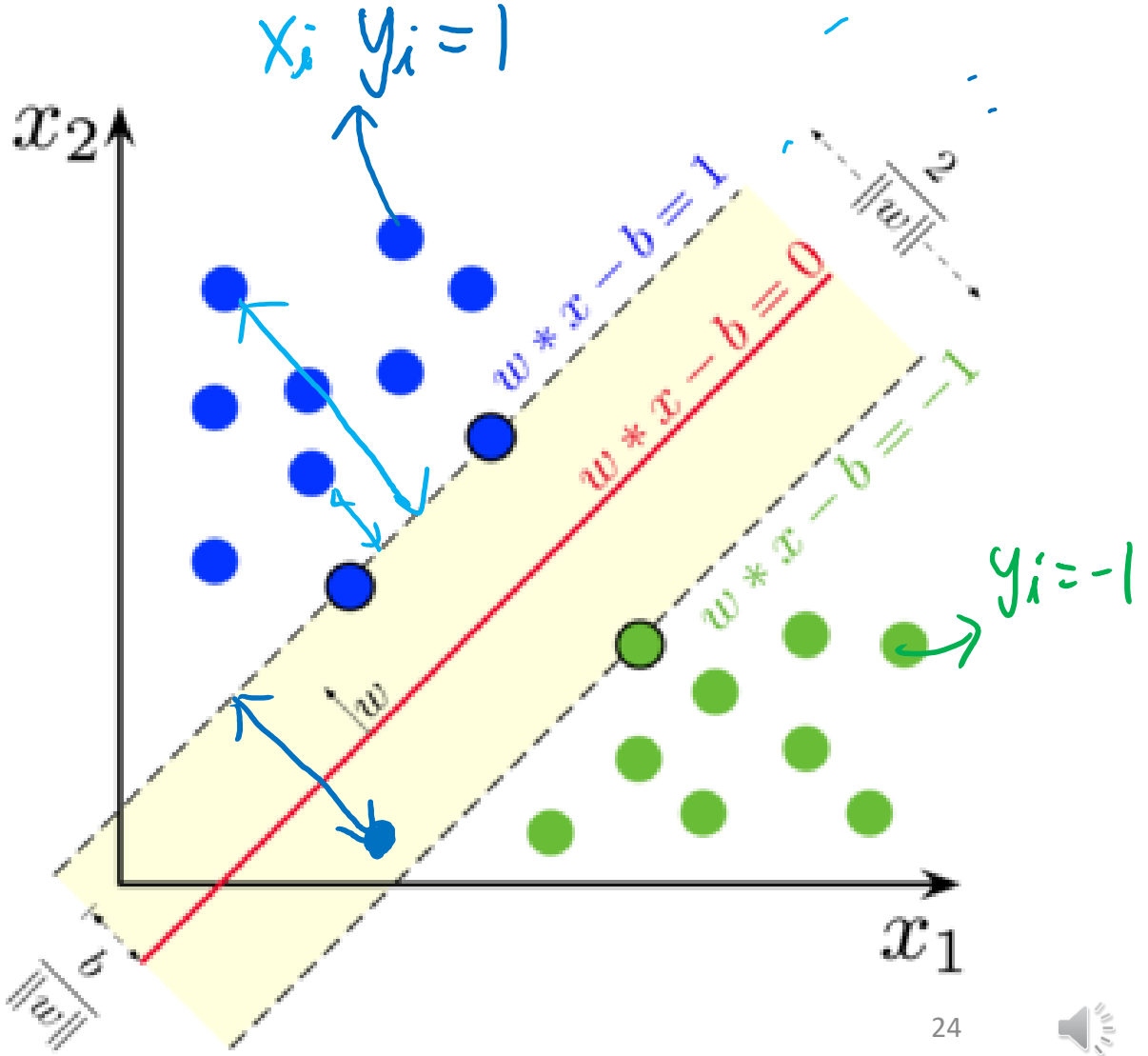
$$y_i' = W^T x_i - b \geq 1$$

$$y_i' = W^T x_i - b \leq -1$$

$$y_i(W^T x_i - b) \geq 1$$

$$\text{Loss} = \max(0, 1 - y_i(W^T x_i - b))$$

Hinge Loss

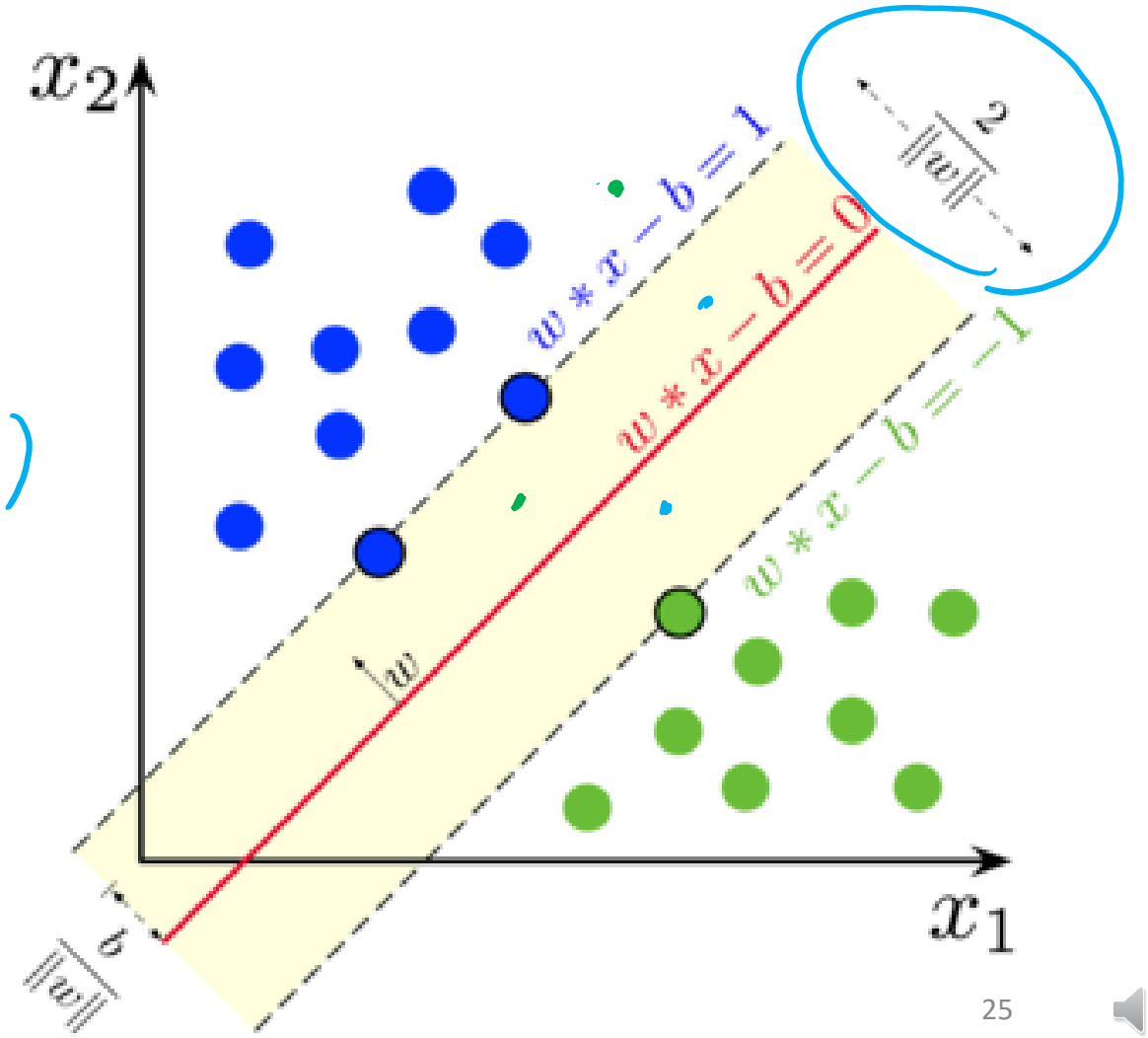


SVM Optimization

- Maximize the margin while reduce hinge loss
- Hinge loss: $\max(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b))$

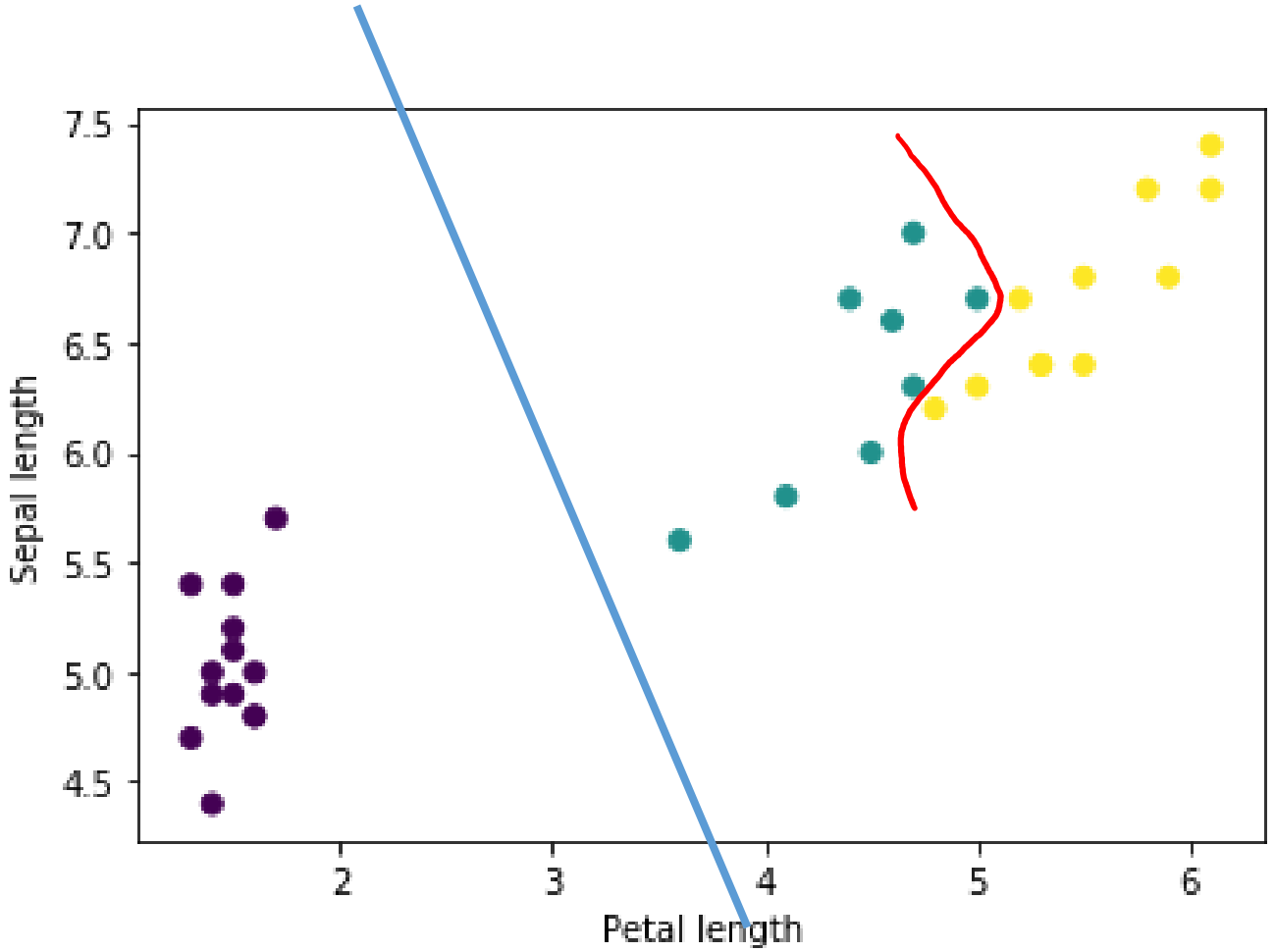
min. $\|w\|$

s.t. $\max(0, 1 - y_i(w^T x_i - b))$



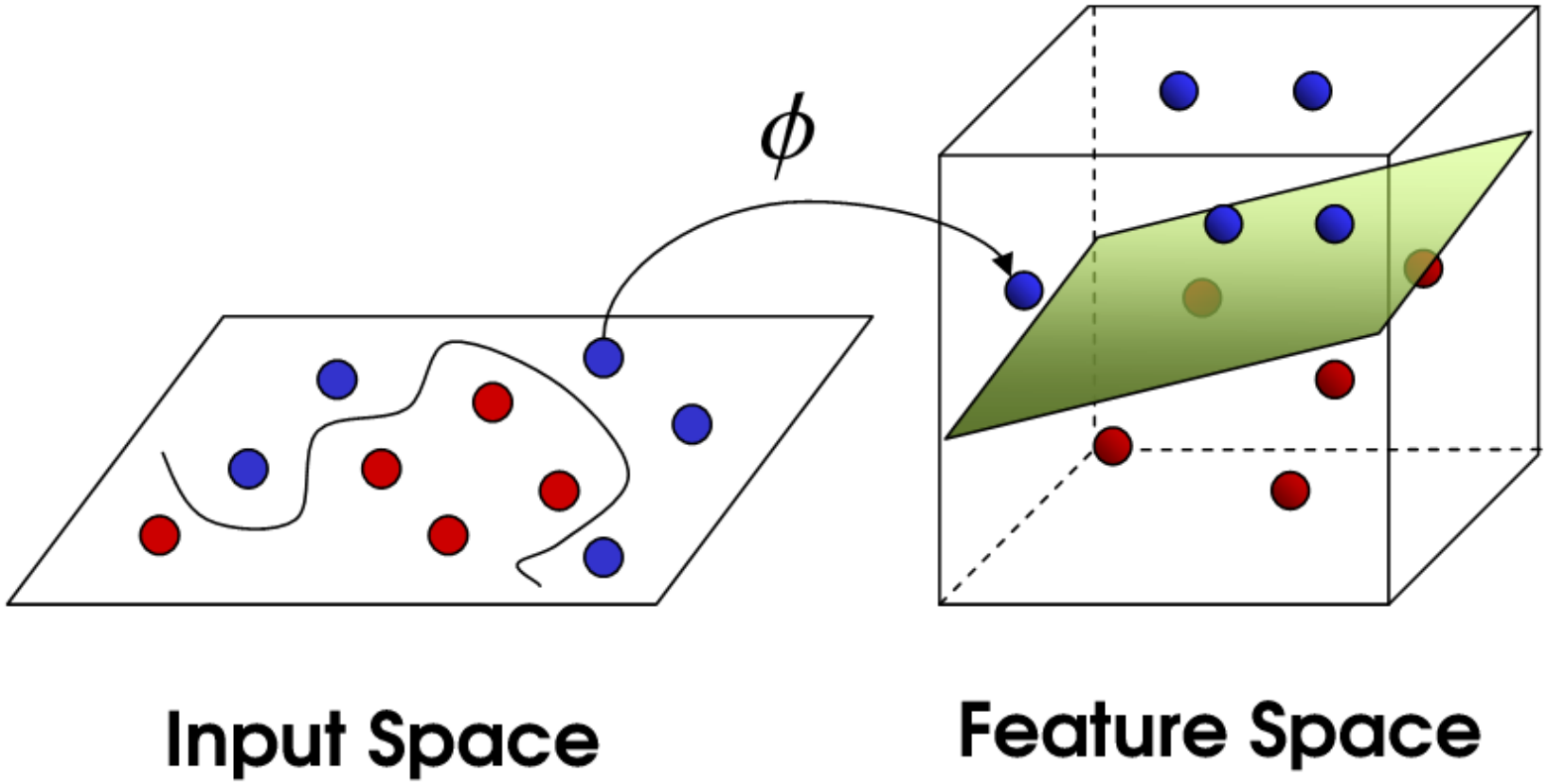
Nonlinear Problem?

- How to separate Versicolor and Virginica?



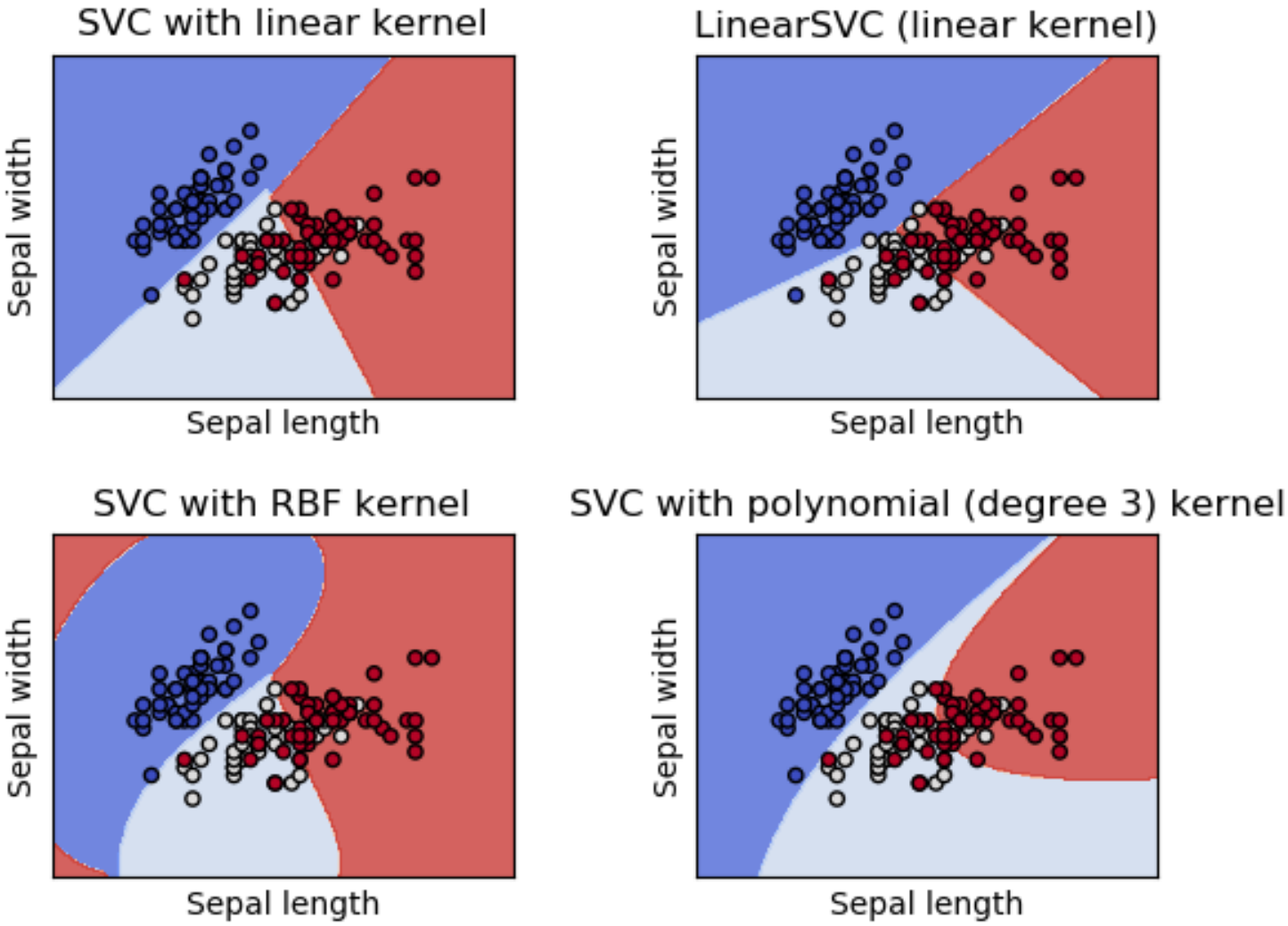
SVM Kernel Trick

- Project data into higher dimension and calculate the inner products

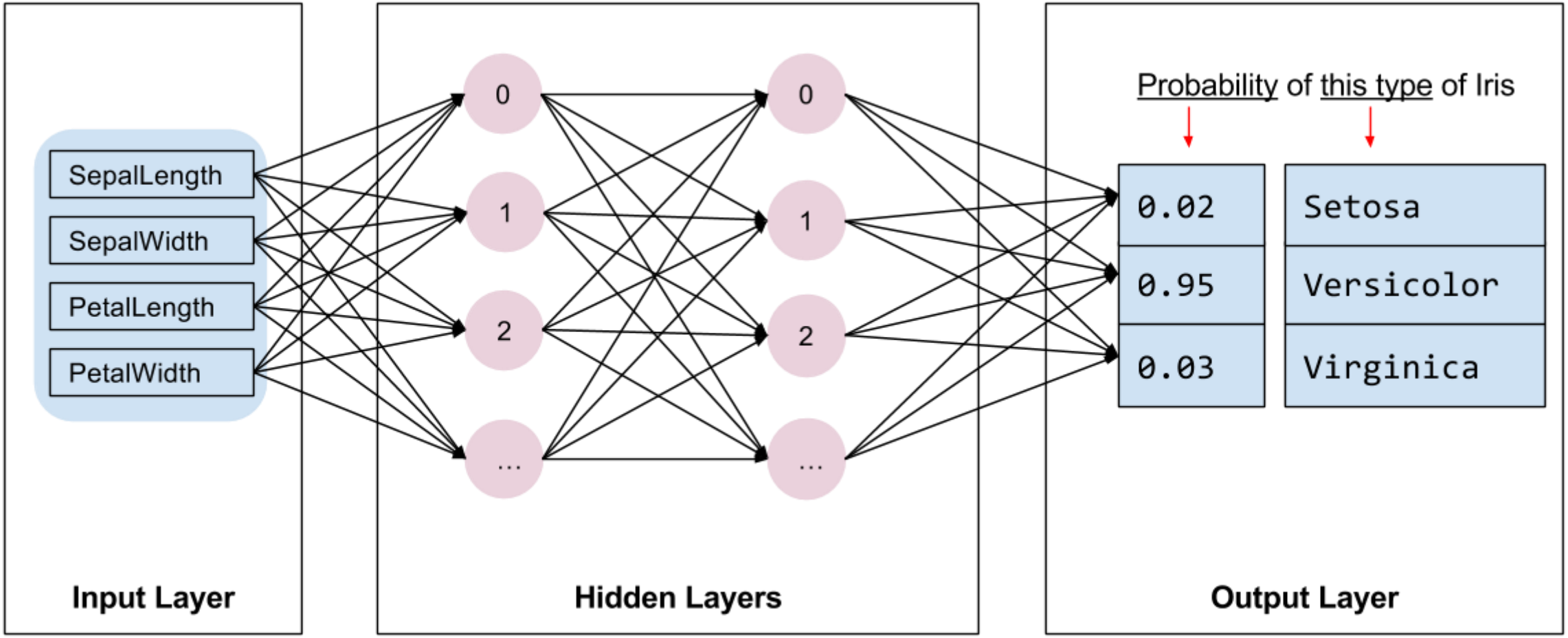


https://datascience.stackexchange.com/questions/17536/kernel-trick-explanation

Nonlinear SVM for Iris Classification

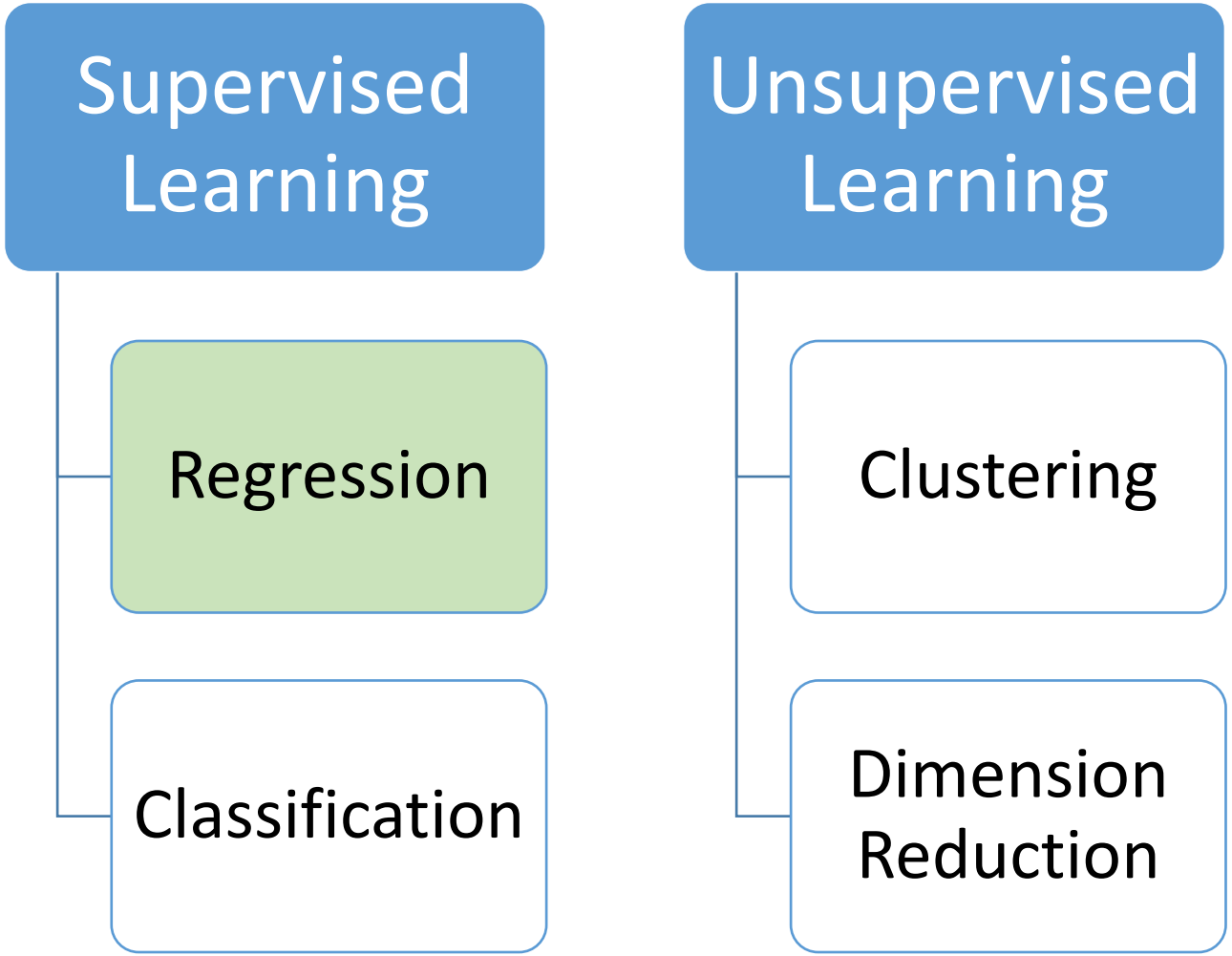


Using Neural Network



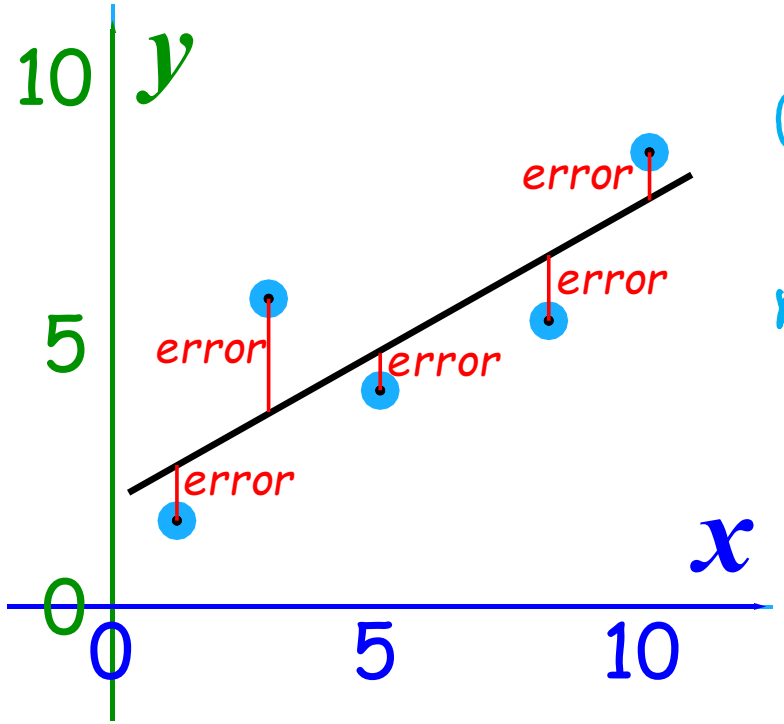
https://www.tensorflow.org/tutorials/customization/custom_training_walkthrough

Supervised and Unsupervised Learning



Linear Regression (Least squares)

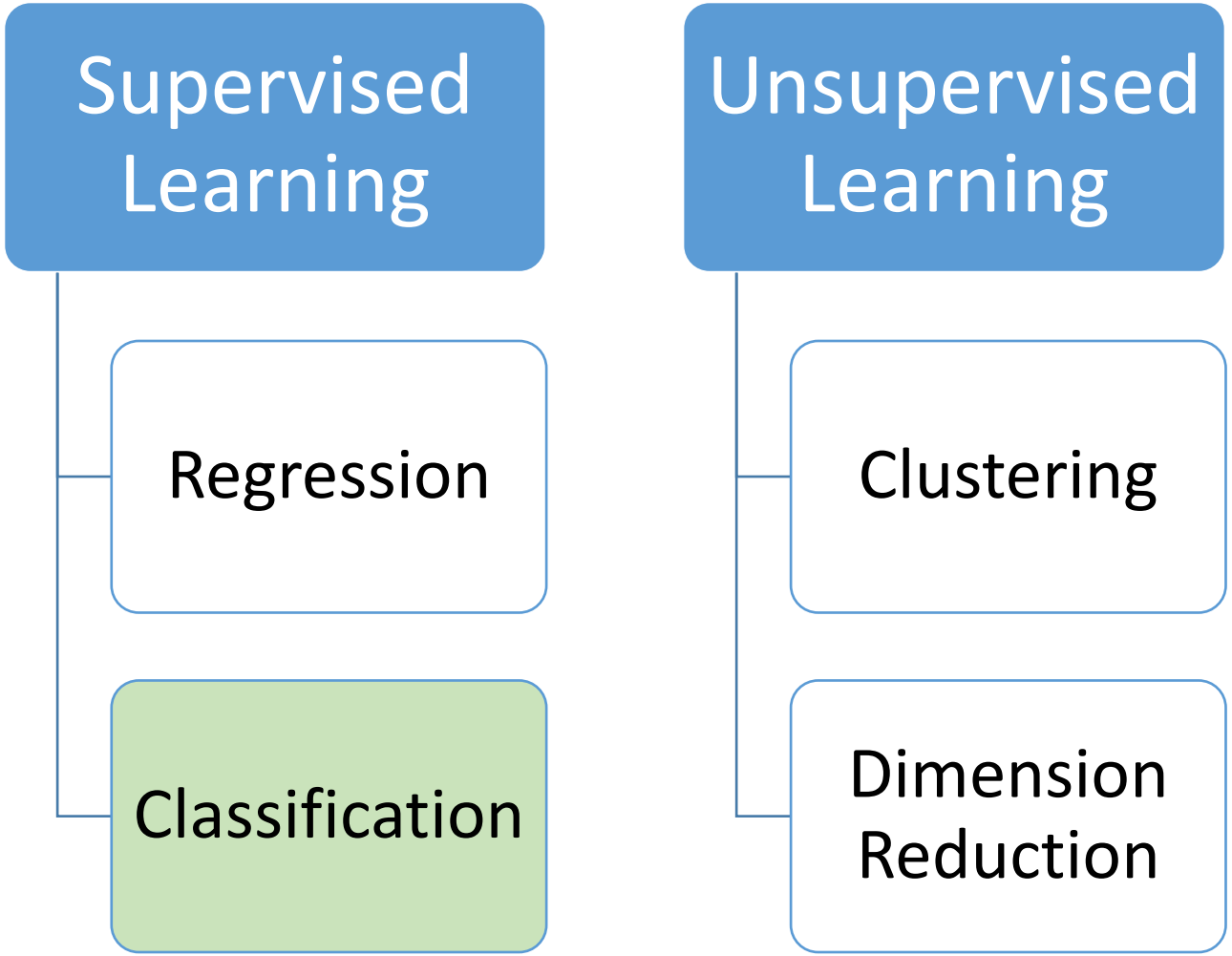
- Find a "line of best fit“ that minimizes the total of the square of the errors



$$y' = wx + b$$

$$\min \sum_{i=0}^{n} (y_i - y_i')^2$$

Supervised and Unsupervised Learning



Logistic Regression

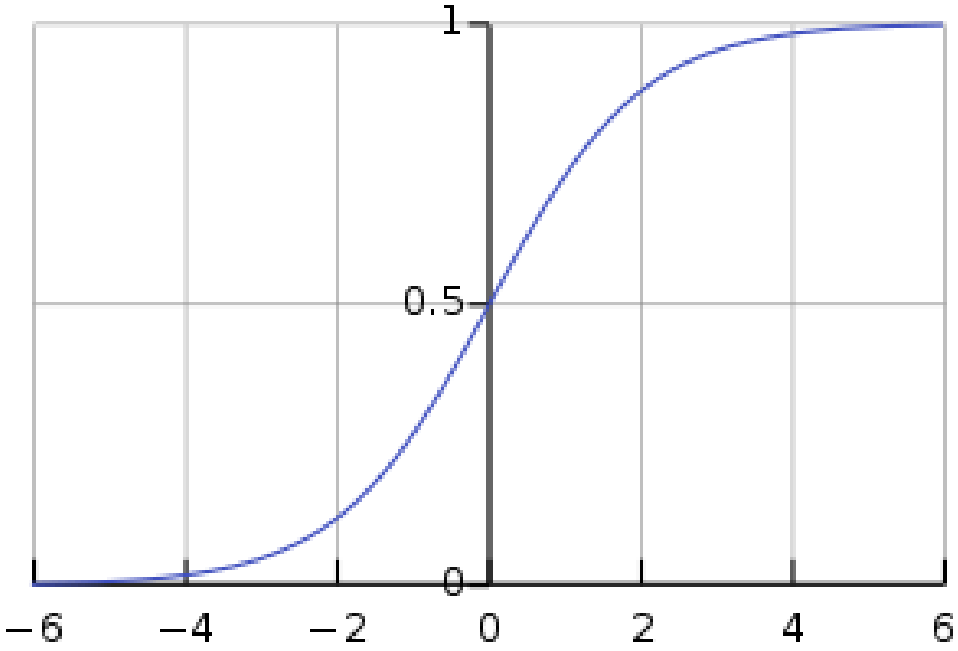
- Sigmoid function

$$S(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}}$$

- Derivative of Sigmoid

$$S(x) = S(x)(1 - S(x))$$

S-shaped curve



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

Decision Boundary

- Binary classification with decision boundary t

$$y' = P(x, w) = P_\theta(x) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x} + b)}}$$

$$y' = \begin{cases} 0, & x < t \\ 1, & x \geq t \end{cases}$$

Cross Entropy Loss

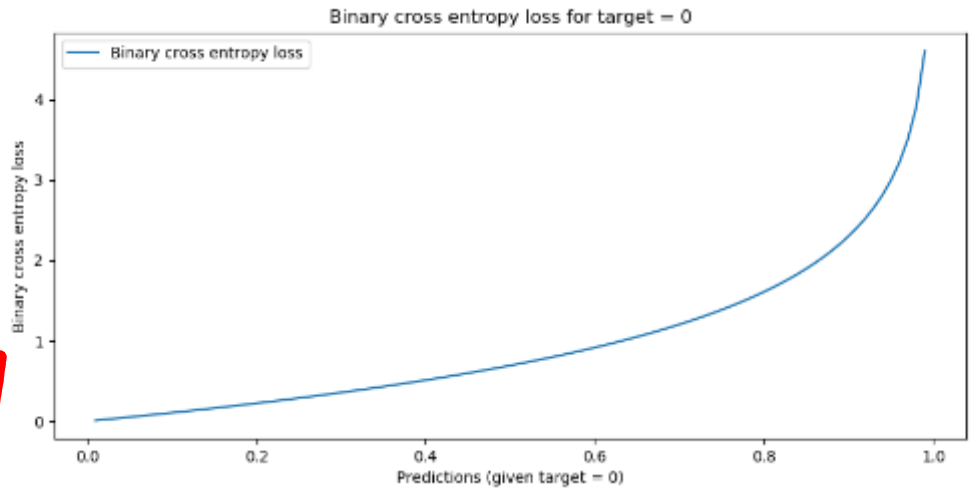
$-\log P(x)$

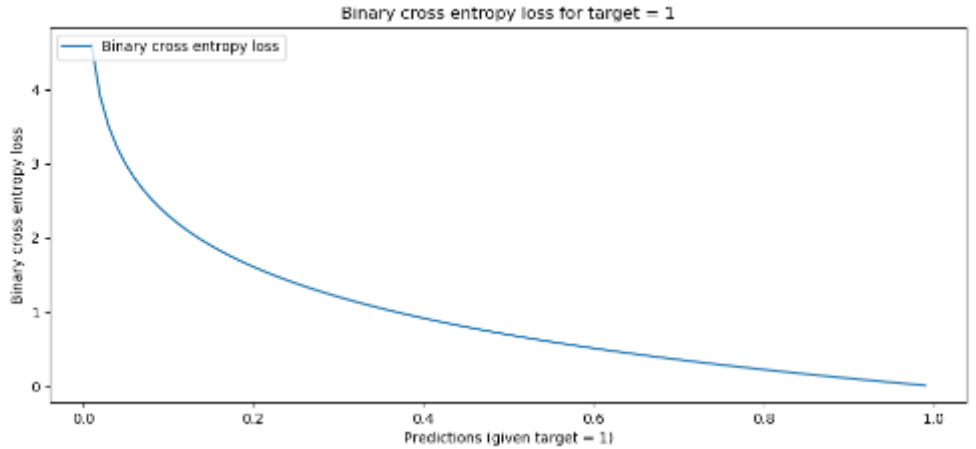
- Loss function: cross entropy

$\log \frac{1}{P(x)}$

information

$$\text{loss}=\begin{cases}-\log\left(1-P_\theta(x)\right), & if\ y=0\\ -\log\left(P_\theta(x)\right), & if\ y=1\end{cases}$$





https://towardsdatascience.com/a-guide-to-neural-network-loss-functions-with-applications-in-keras-3a3baa9f71c5

Cross Entropy Loss

- Loss function: cross entropy

$$\text{loss}=\begin{cases}-\log\big(1-P_\theta(x)\big), & if\ y=0\\ -\log\big(P_\theta(x)\big), & if\ y=1\end{cases}$$

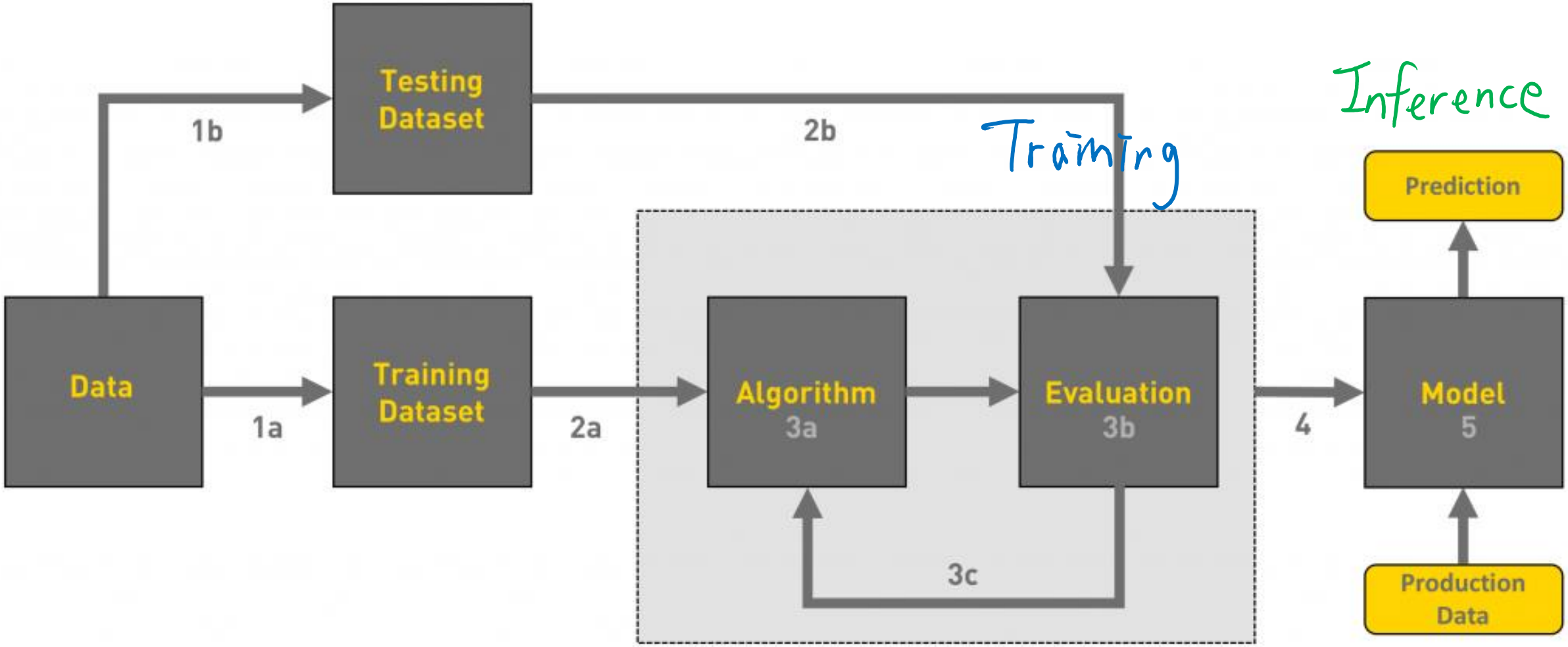
$$\Rightarrow L_\theta(\mathrm{x})=-y\log\big(P_\theta(x)\big)+-(1-\mathrm{y})\log\big(1-P_\theta(x)\big)$$

$$\nabla L_W(\mathrm{x})=-\big(y-P_\theta(x)\big)x$$

y'

https://towardsdatascience.com/a-guide-to-neural-network-loss-functions-with-applications-in-keras-3a3baa9f71c5

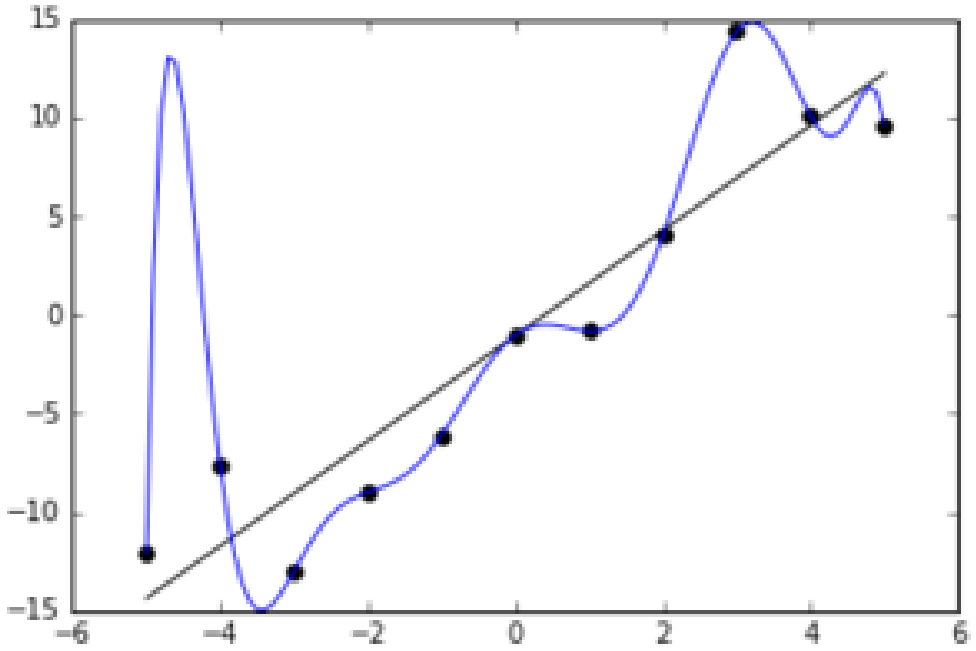
Machine Learning Workflow



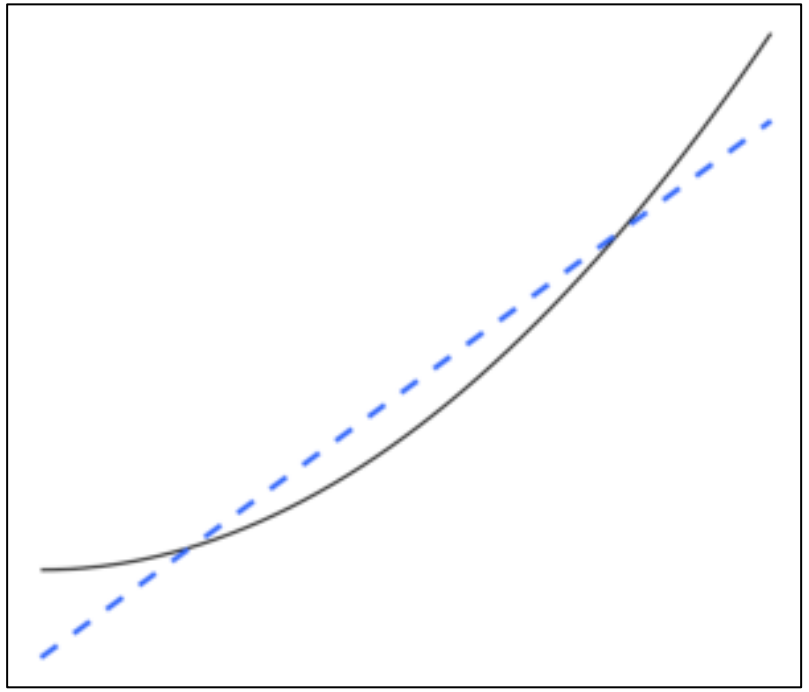
https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94

Overfitting and Underfitting

Overfitting



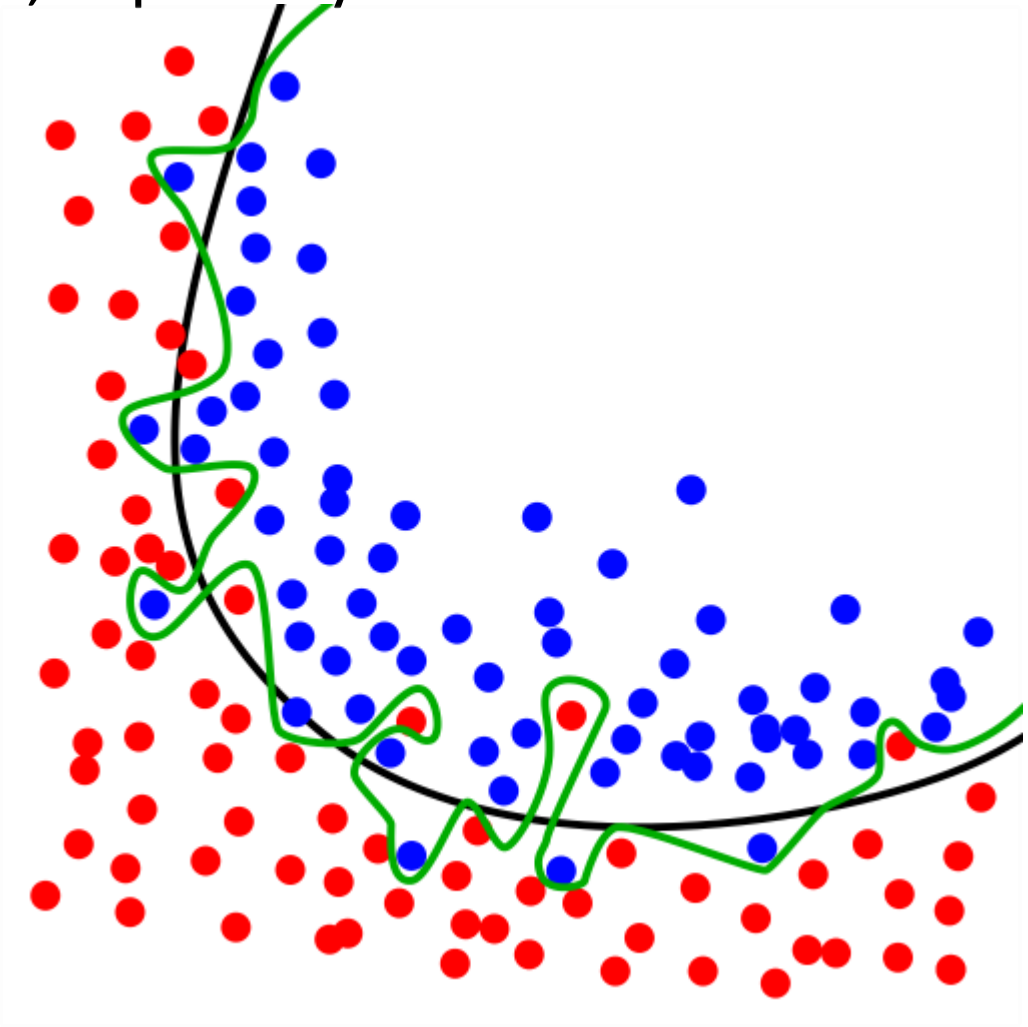
Underfitting



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

Overfitting (以偏概全)

- Overfitting is common, especially for neural networks

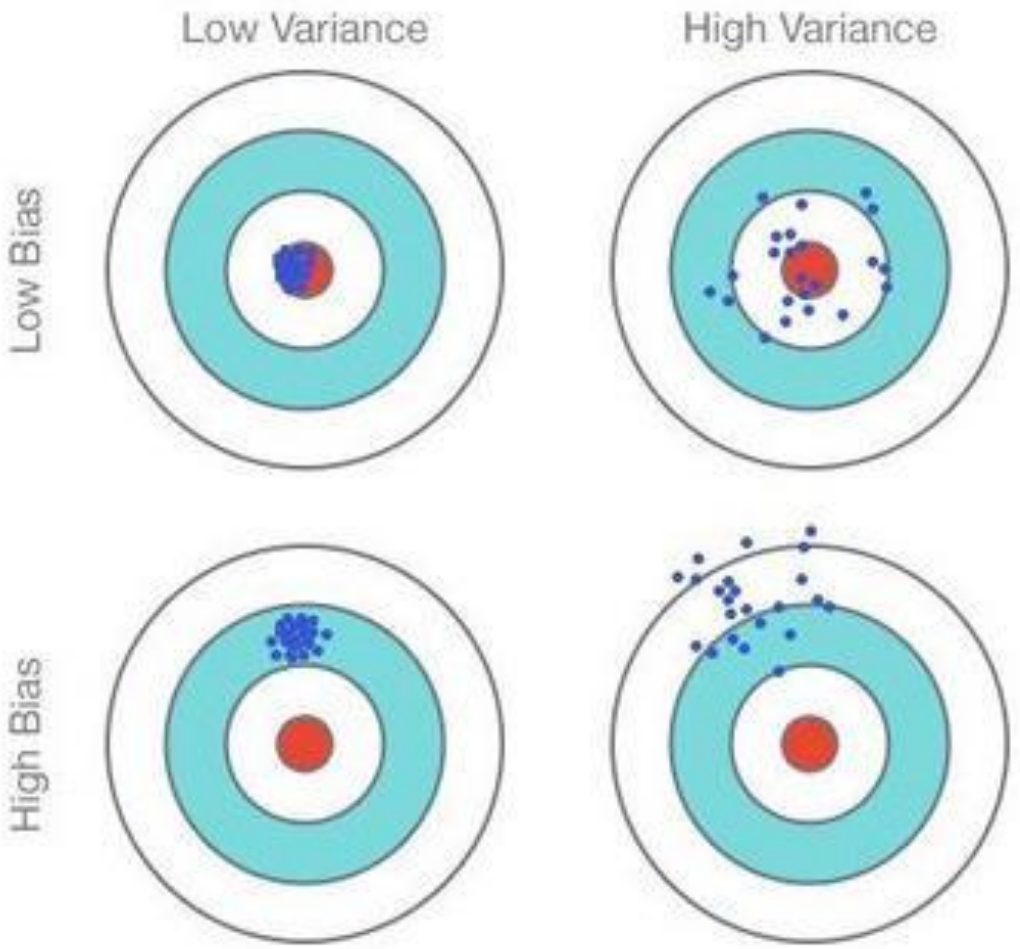


Neural Network Urban Legend: Detecting Tanks

- Detector learned the illumination of photos

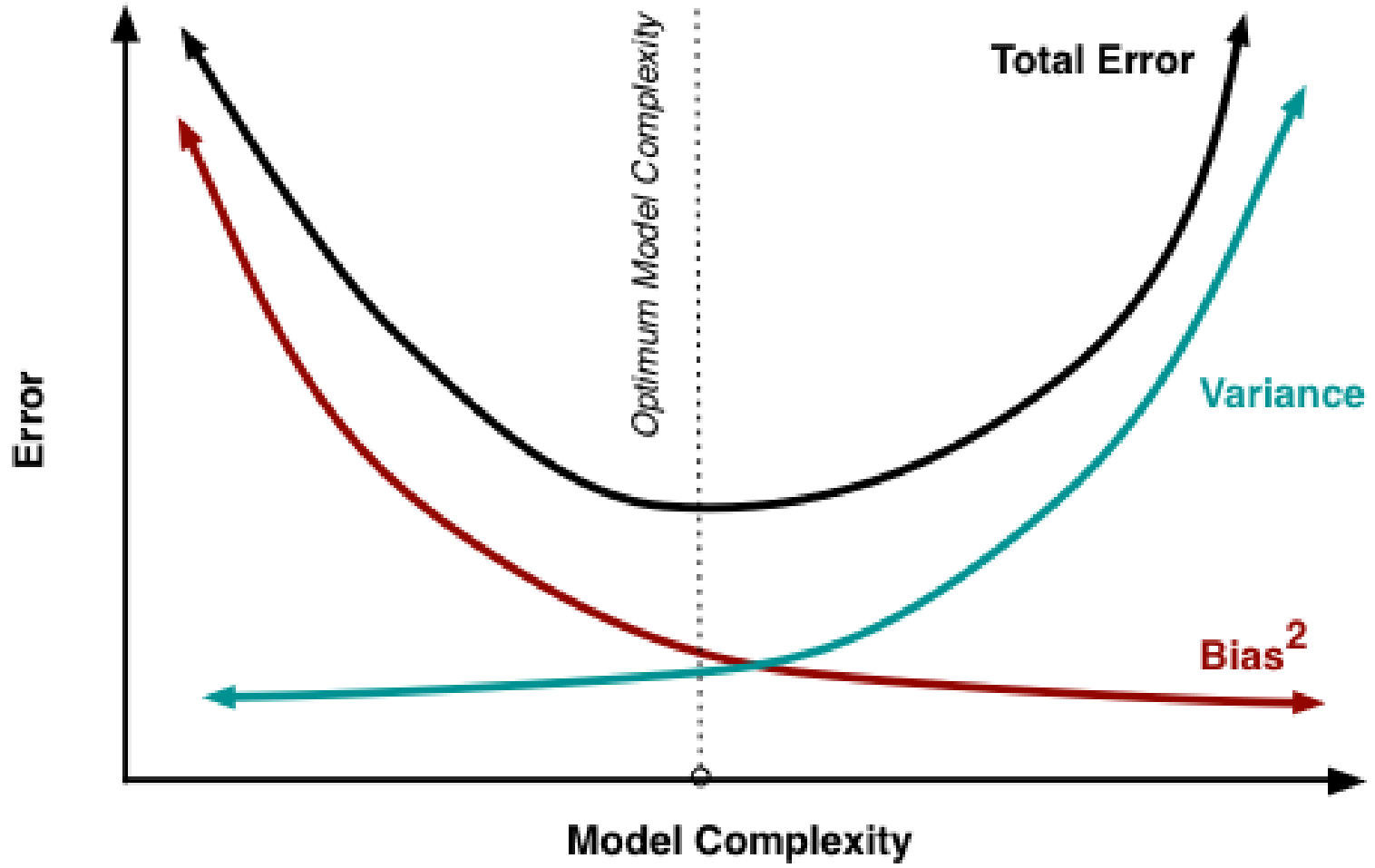
Bias and Variance Trade-off

- Model with high variance overfits to training data and does not generalize on unseen test data



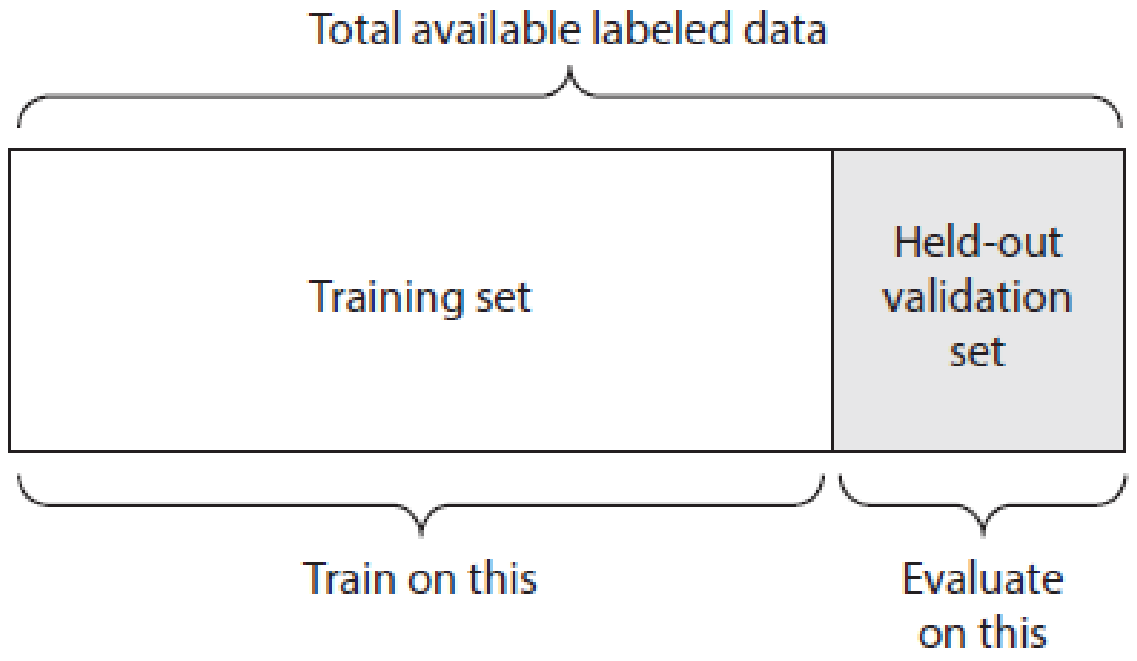
http://scott.fortmann-roe.com/docs/BiasVariance.html

Model Selection



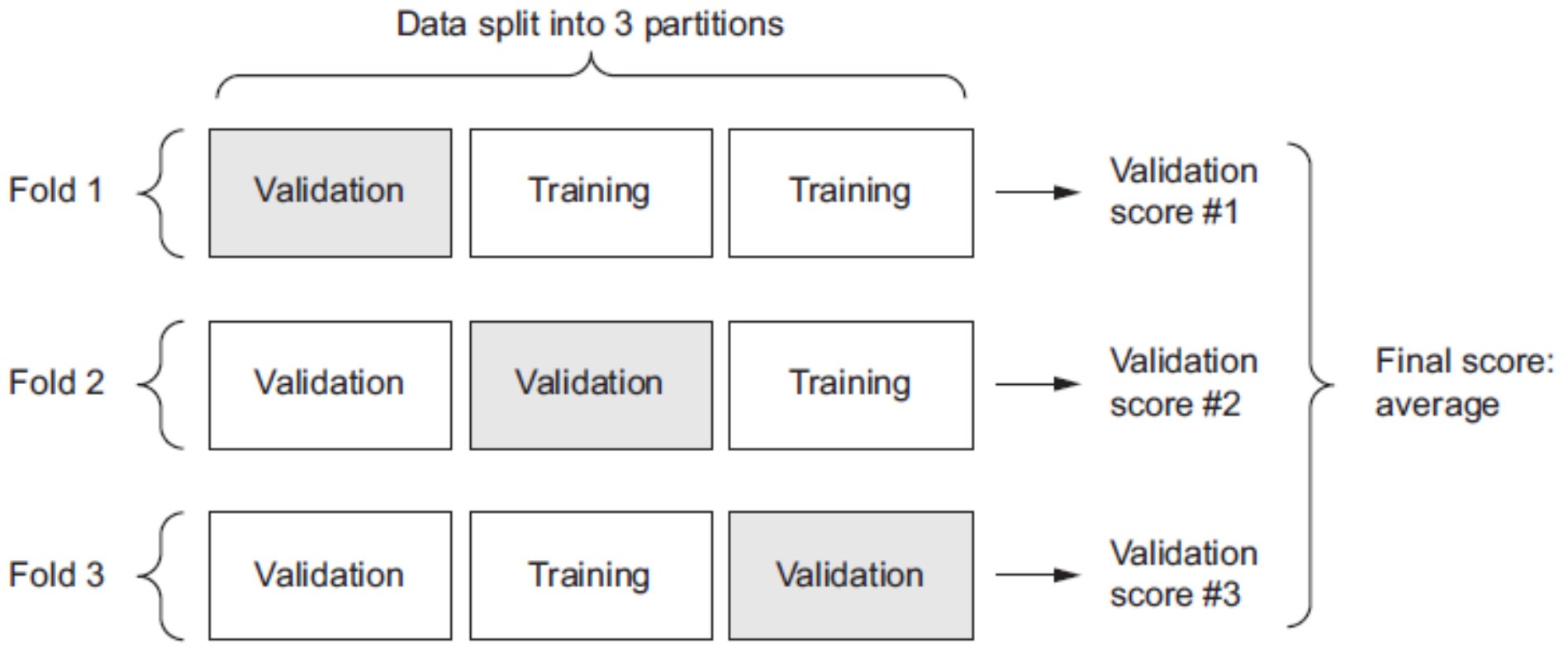
Training, Validation, Testing

- Never leak test data information into our model
- Tuning the *hyperparameters* of our model on validation dataset



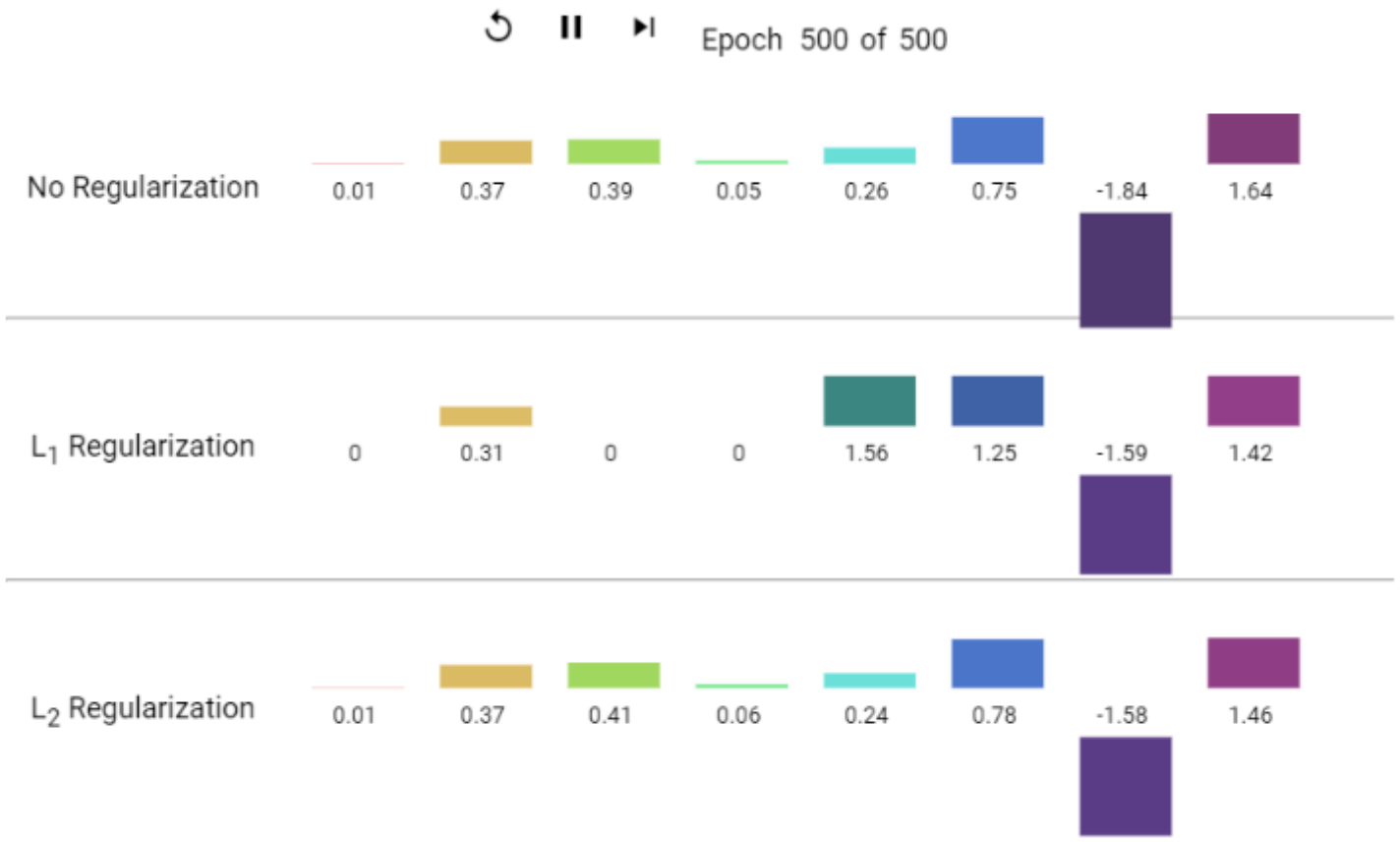
K-Fold Cross Validation

- Lower the variance of validation set



Regularization

- https://developers.google.com/machine-learning/crash-course/regularization-for-sparsity/l1-regularization



Metrics:
Accuracy vs. Precision
in Binary Classification

Confusion Matrix

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

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

Confusion Matrix

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

		True condition			
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	F_1 score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Popular Metrics

- Notations
  - P: positive samples, N: negative samples, P’: predicted positive samples, TP: true positives, TN: true negatives
- Recall = $\frac{\mathrm{TP}}{\mathrm{P}}$
- Precision = $\frac{\mathrm{TP}}{\mathrm{P}'}$
- Accuracy = $\frac{\mathrm{TP}+\mathrm{TN}}{P+\mathrm{N}}$
- F1 score = $\frac{2}{\frac{1}{recall}+\frac{1}{precision}}$
- Miss rate = false negative rate = 1 – recall

Coronavirus Example

- Precision = 8 / 18 = 44%
- Accuracy = (8 + 90) / 110 = 89%

數感實驗室 NUMERACY LAB

一種錯誤率，各自表述!?

這幾天有新聞報導，捷克使用中國製的快篩劑來檢驗新冠病毒
結果錯誤率高達 80%！

	真的生病	健康	
檢測陽性	8	10	18
檢測陰性	2	90	92
	10	100	

誤判健康者生病 ≠ 沒檢查出患者

假警報 (false alarm)　失誤 (miss)

https://www.facebook.com/numeracylab/posts/2997362375951453

Accuracy

- Example: Classifying tumors as malignant or benign (Google Machine Learning Crash Course)

True Positive (TP): • Reality: Malignant • ML model predicted: Malignant • **Number of TP results: 1**	**False Positive (FP):** • Reality: Benign • ML model predicted: Malignant • **Number of FP results: 1**
False Negative (FN): • Reality: Malignant • ML model predicted: Benign • **Number of FN results: 8**	**True Negative (TN):** • Reality: Benign • ML model predicted: Benign • **Number of TN results: 90**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 90}{1 + 90 + 1 + 8} = 0.91$$

https://developers.google.com/machine-learning/crash-course/classification/accuracy

Precision and Recall

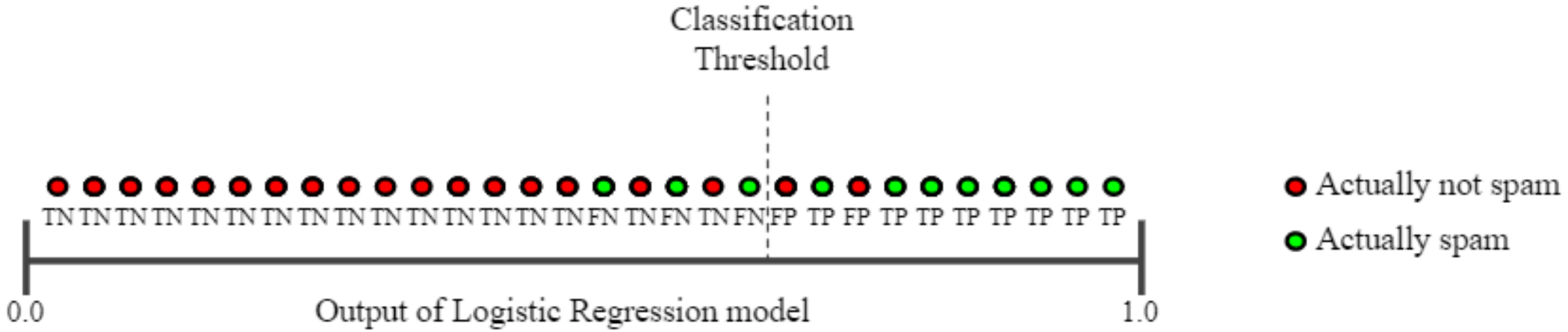
- Classifying tumors as malignant or benign

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{1}{1+1} = 0.5$$

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{1}{1+8} = 0.11$$



https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

ROC Curve

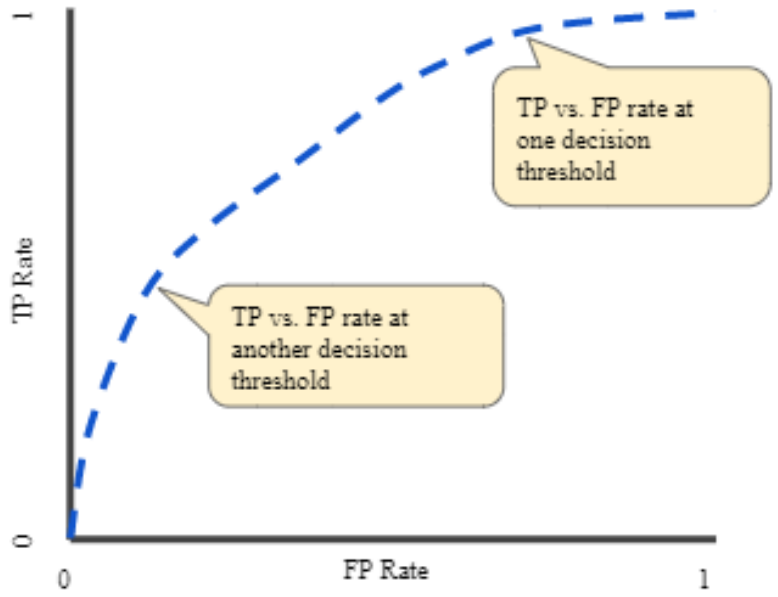
- ROC (Receiver Operating Characteristic) Curve

True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (**FPR**) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$



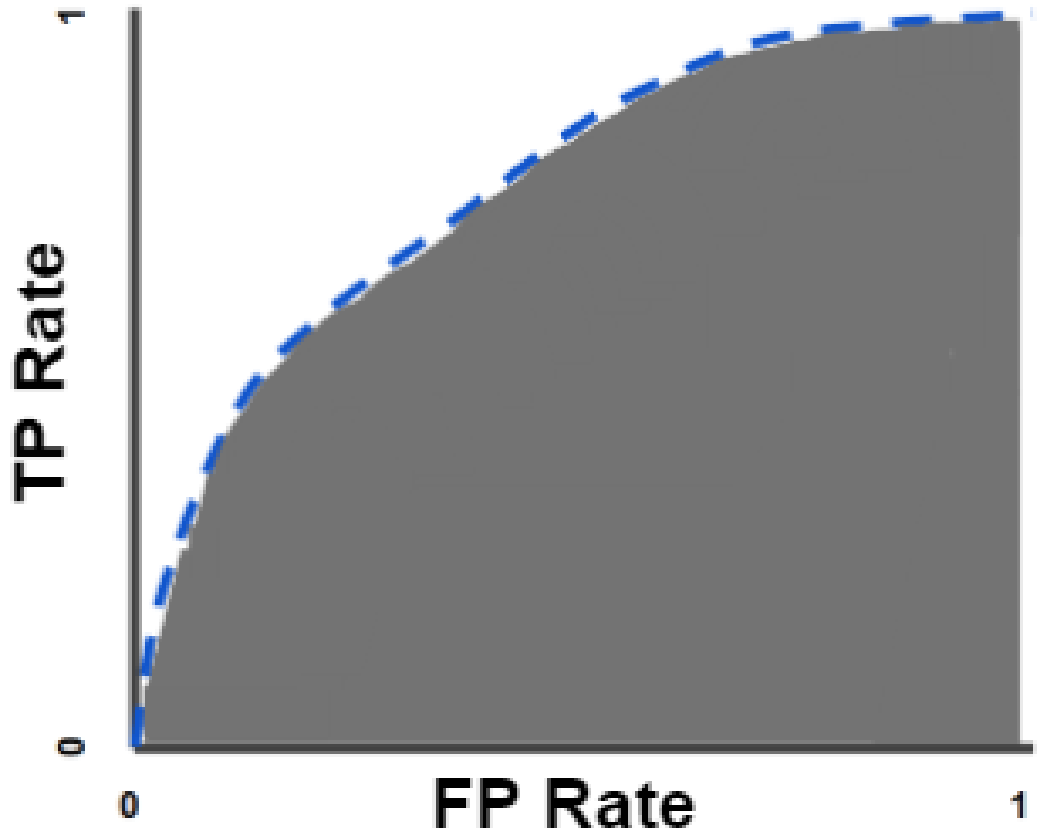
https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

https://bwfinsight.blog/2018/05/10/the-link-between-world-war-ii-and-your-predictive-models/

AUC (Area under the ROC Curve)

- AUC ranges in value from 0 to 1.
  - AUC of 0.0 = predictions are 100% wrong (X)
  - AUC of 1.0 = predictions are 100% correct (O)

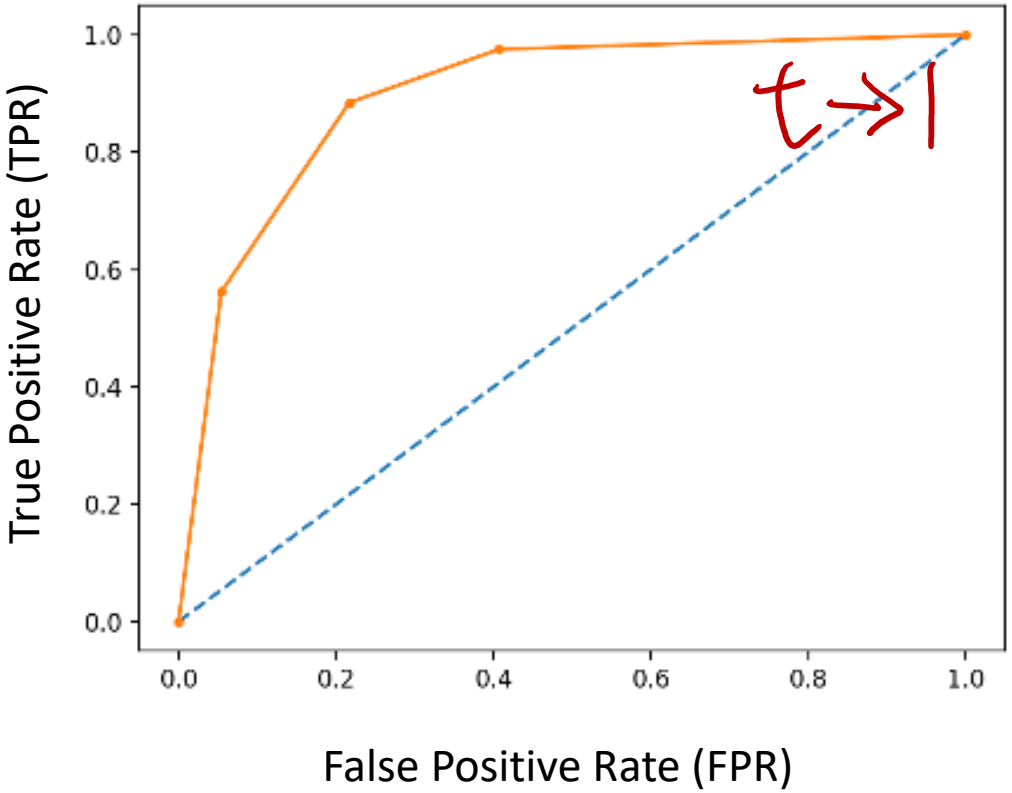
- Advantages
  - Scale-invariant
  - Classification-threshold-invariant
    - Not suitable for email spam detection



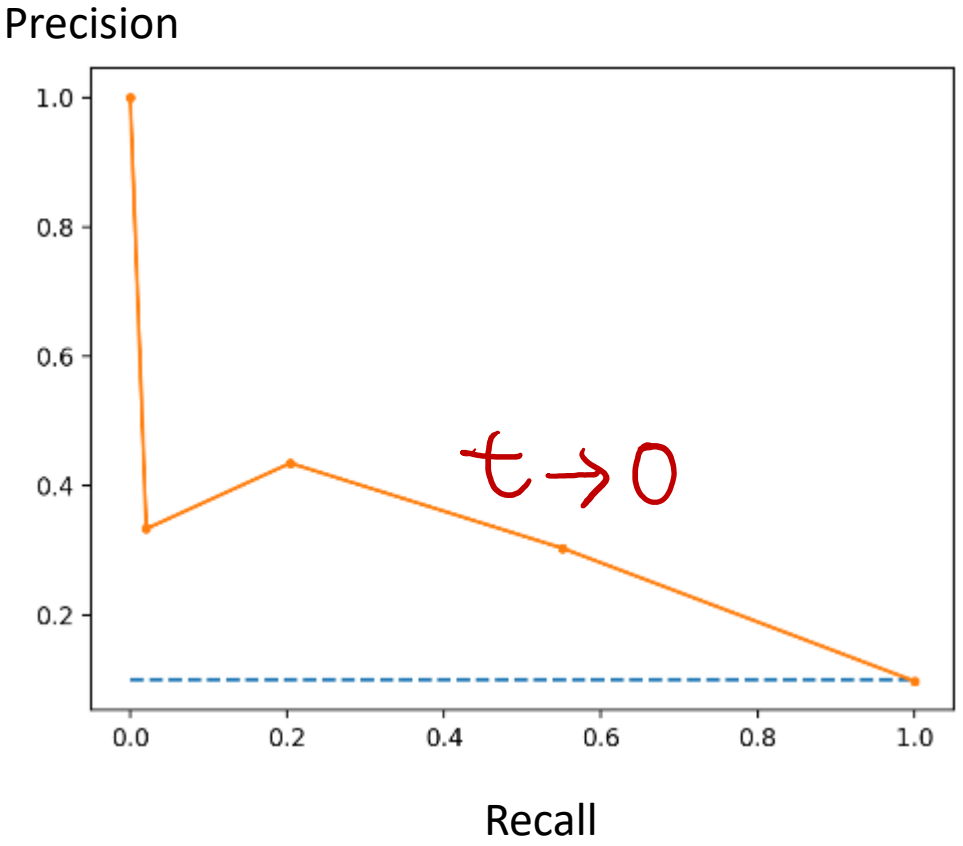
https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Evaluate Decision Boundary *t*

- ROC (Receiver Operating Characteristic) Curve



- Precision-Recall (PR) Curve



Summary of ML Training Flow

1. Defining the problem and assembling a dataset
2. Choosing a measure of success
3. Deciding on an evaluation protocol
4. Preparing your data
5. Developing a model that does better than a baseline
6. Scaling up: developing a model that overfits
7. Regularizing your model and tuning your hyperparameters

Pedro Domingos – Things to Know about Machine Learning

Useful Things to Know about Machine Learning

1. It's generalization that counts
2. Data alone is not enough
3. Overfitting has many faces
4. Intuition fails in high dimensions
5. Theoretical guarantees are not what they seem
6. More data beats a cleverer algorithm
7. Learn many models, not just one

Pedro Domingos, "A Few Useful Things to Know about Machine Learning," Commun. ACM, 2012

It’s Generalization that Counts

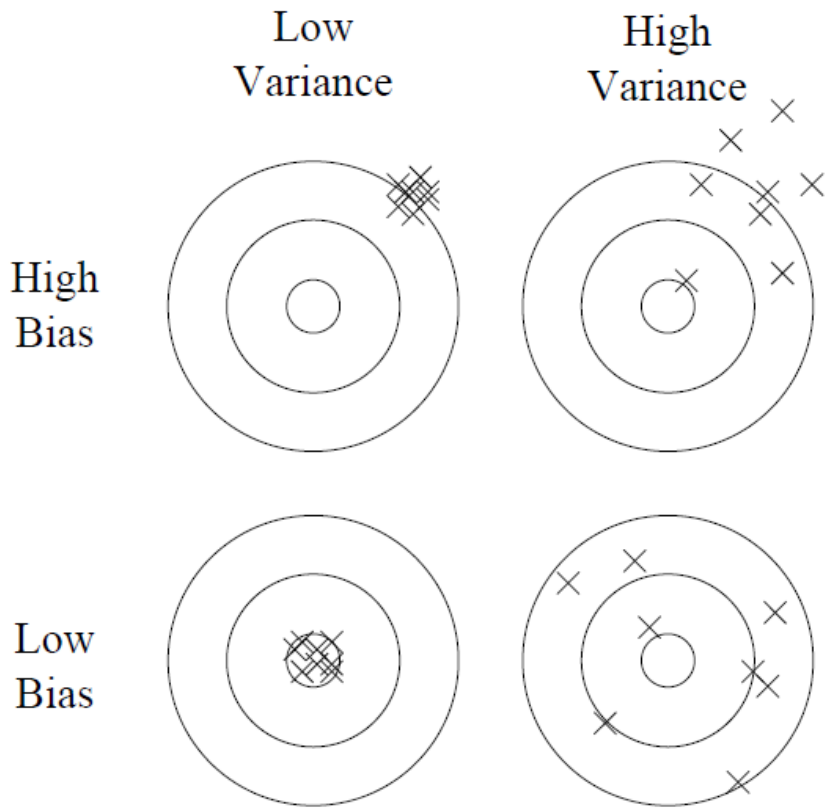
- The goal of machine learning is to *generalize* beyond the examples in the training set
- Don’t use test data for training
- Use cross validation to verify your model

Data Alone Is Not Enough

- No free lunch theorem (Wolpert)
  - Every learner must embody some knowledge or assumptions beyond the data
- Learners combine knowledge with data to grow programs

Overfitting Has Many Faces

- Ex: when your model accuracy is 100% on training data but only 50% on test data, when in fact it could have 75% on both, it has overfit.
- Overfitting has many forms. Example: bias & variance
- Combat overfitting
  - Cross validation
  - Add regularization term



Intuition Fails in High Dimensions (Number of Features)

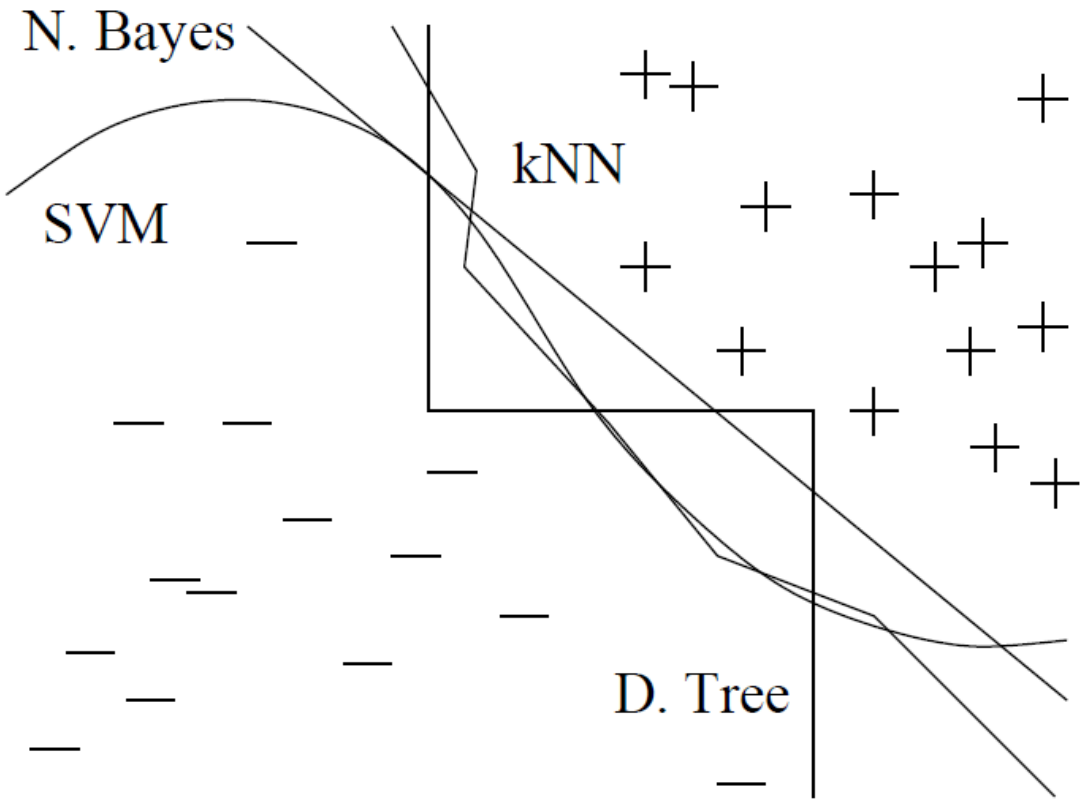
- Curse of Dimensionality
- Algorithms that work fine in low dimensions fail when the input is high-dimensional
- Generalizing correctly becomes exponentially harder as the dimensionality of the examples grows
- Our intuition only comes from 3-dimension

Theoretical Guarantees Are Not What They Seem

- Theoretical bounds are usually very loose
- The main role of theoretical guarantees in machine learning is to help understand and drive force for algorithm design

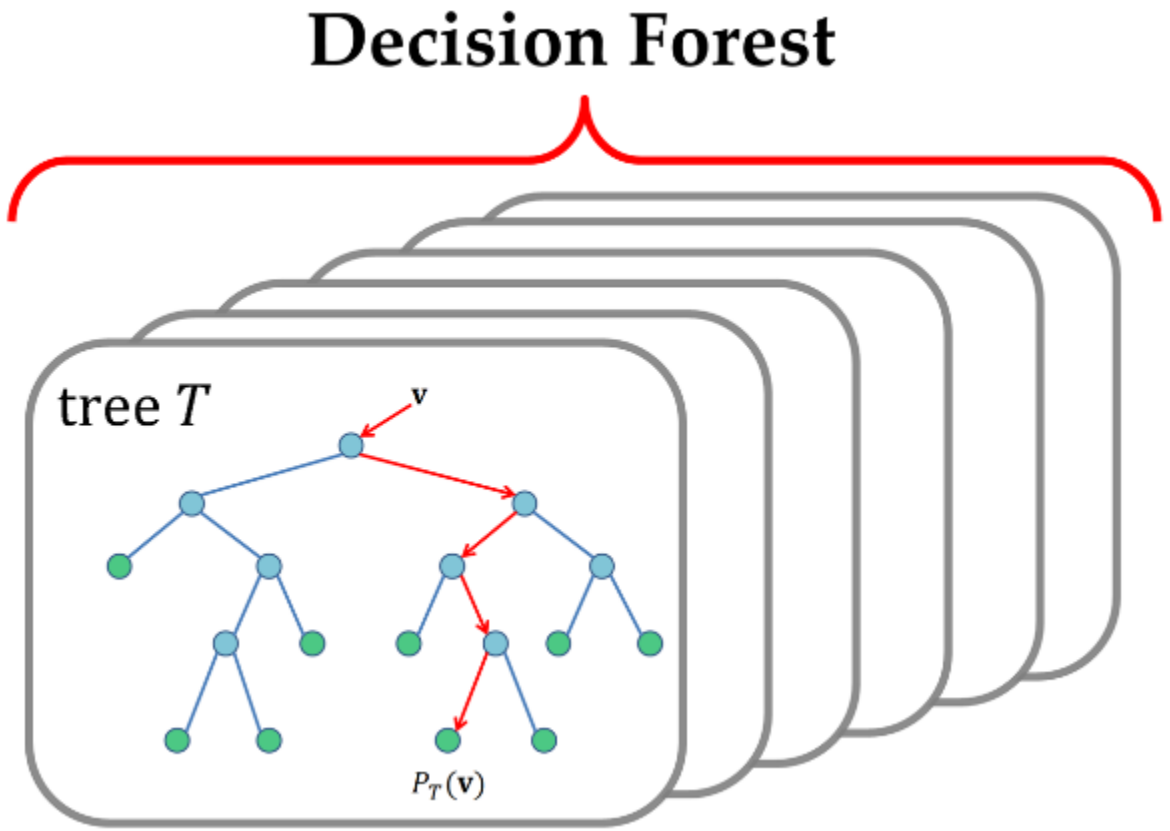
More Data Beats a Cleverer Algorithm

- Try simplest algorithm first



Learn Many Models, Not Just One

- Ensembling methods: Random Forest ,XGBoost, Late Fusion
- Combining different models can get better results



References

- Francois Chollet, “Deep Learning with Python”, Chapter 4
- Pedro Domingos, “A Few Useful Things to Know about Machine Learning,” Commun. ACM, 2012
- https://ml-cheatsheet.readthedocs.io/en/latest/index.html
- https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/
- https://towardsdatascience.com/data-types-in-statistics-347e152e8bee
- https://en.wikipedia.org/wiki/Naive_Bayes_classifier