

# STATISTICS FOR NON-BIOSTATISTICIAN

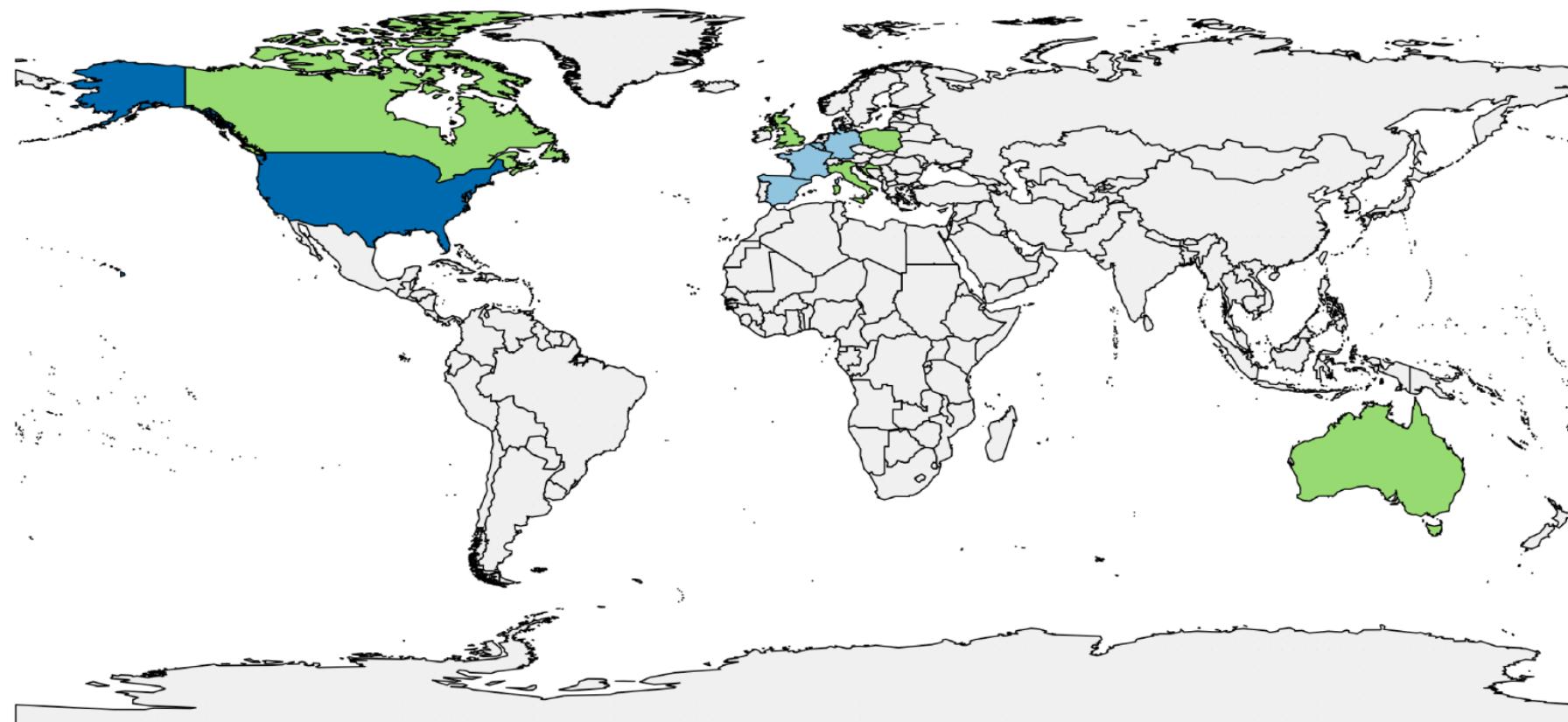
*What I need to know  
when I analyze my biological experiments*

**Tiphaine Martin, M.Eg., Ph.D.**  
**9/8/2020**

# Bioinformatician/Biostatistician - Multi-omic/EHR

TIPHANE.MARTIN@MSSM.EDU

9/8/20



- Postdoc - Multi-omic/EHR/human – liquid biopsy/BioMe biobank, microbiome
- PhD - Multi-omic/EHR/human – autoimmune thyroid diseases/rare diseases (hematology) – CHARGE consortium (epigenome)/EpiTwin/Blueprint Project/Interval Project/UK10K/TwinsUK/Genomics England
- Research Engineer – Comparative genomics/yeast – Marine genomics – AI/robotics – human/virus EGI/ELIXIR/Genolevures/ BioMerieux/It-omics

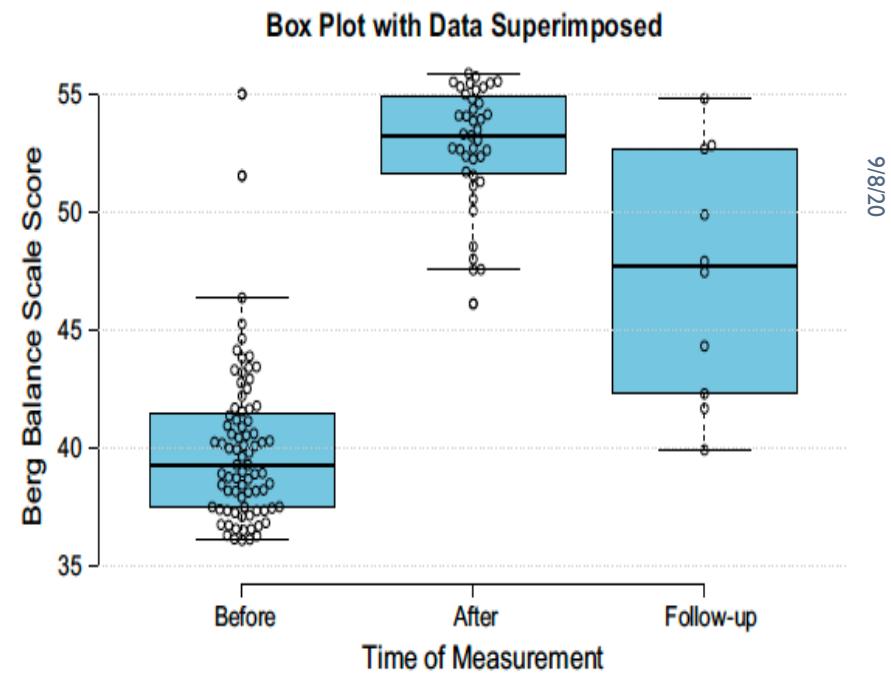
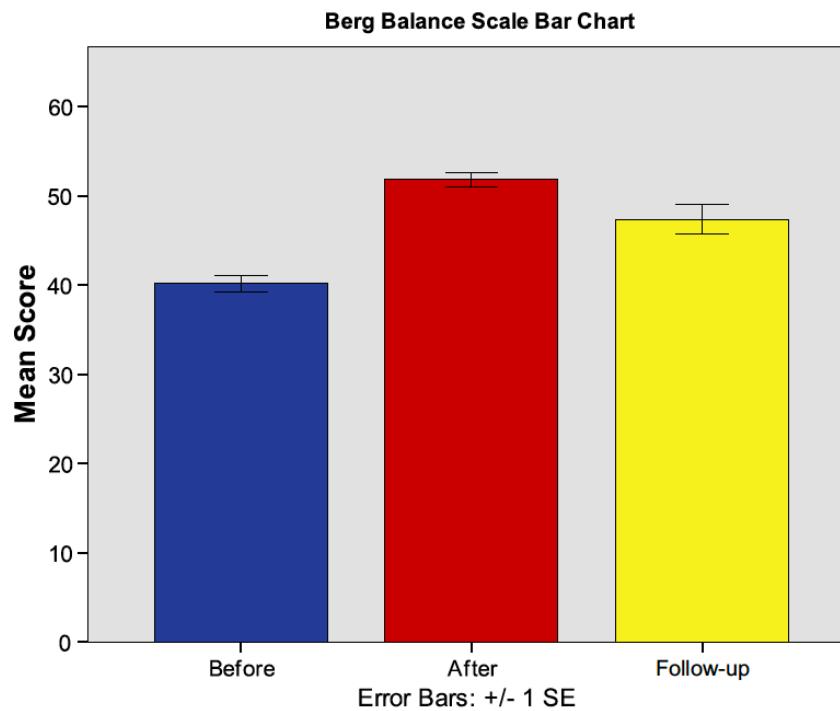
# OUTLINE

- Visualization
- Basic statistics
- Artificial Intelligence vs Machine learning
- Examples

# VISUALIZATION

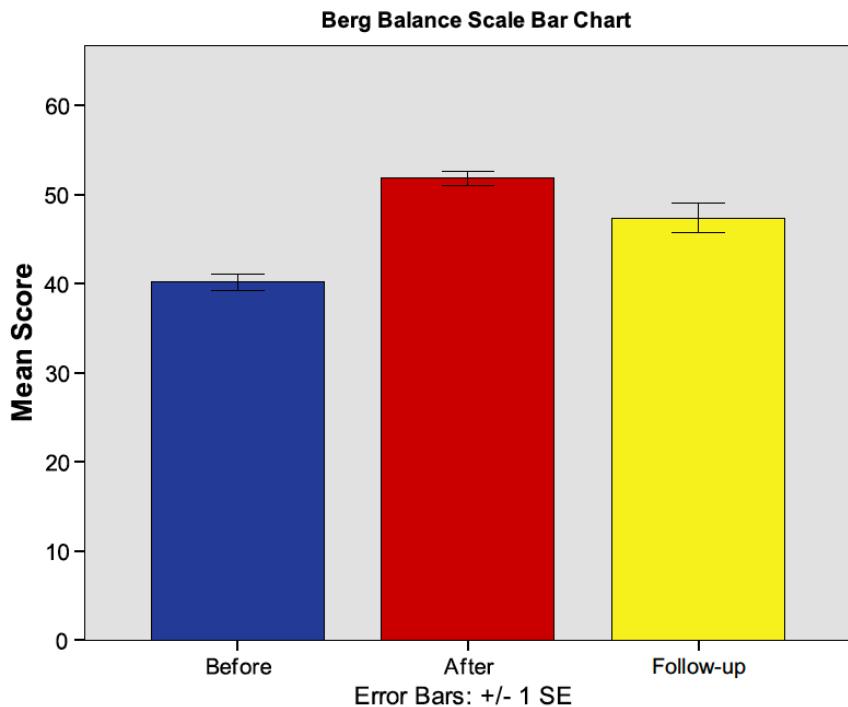
# VISUALIZATION

Which one do you prefer? And why ?

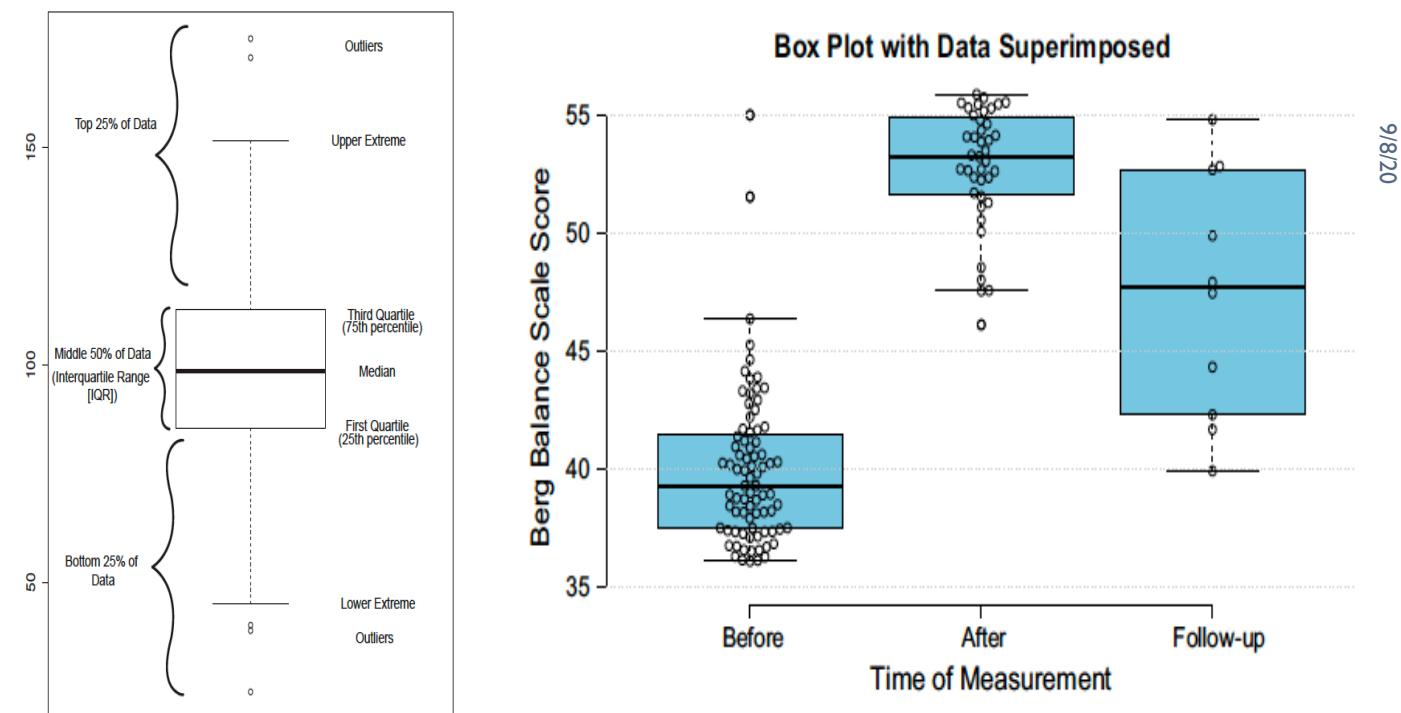


# VISUALIZATION

To avoid

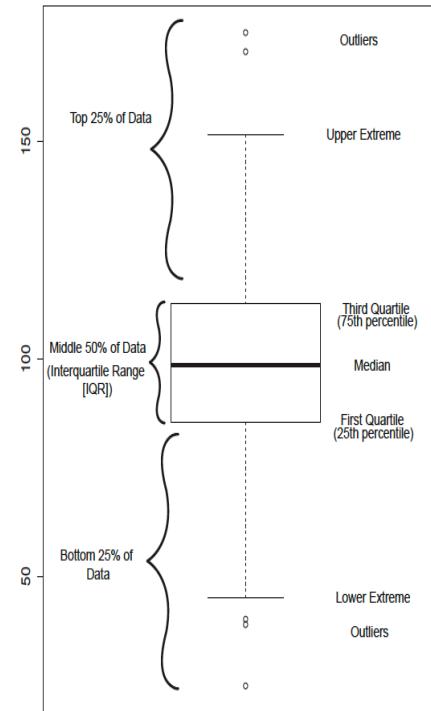
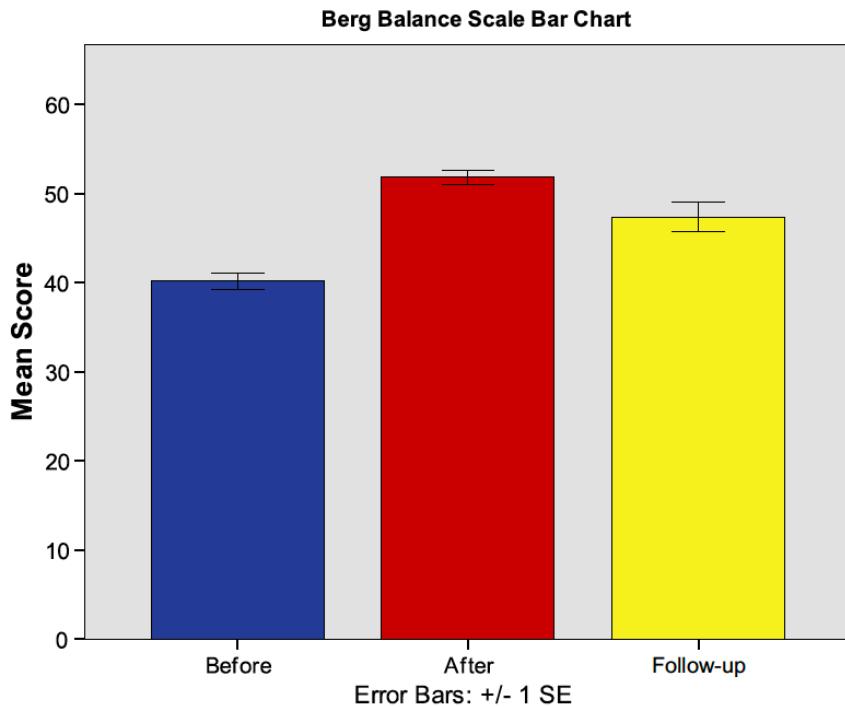


Better

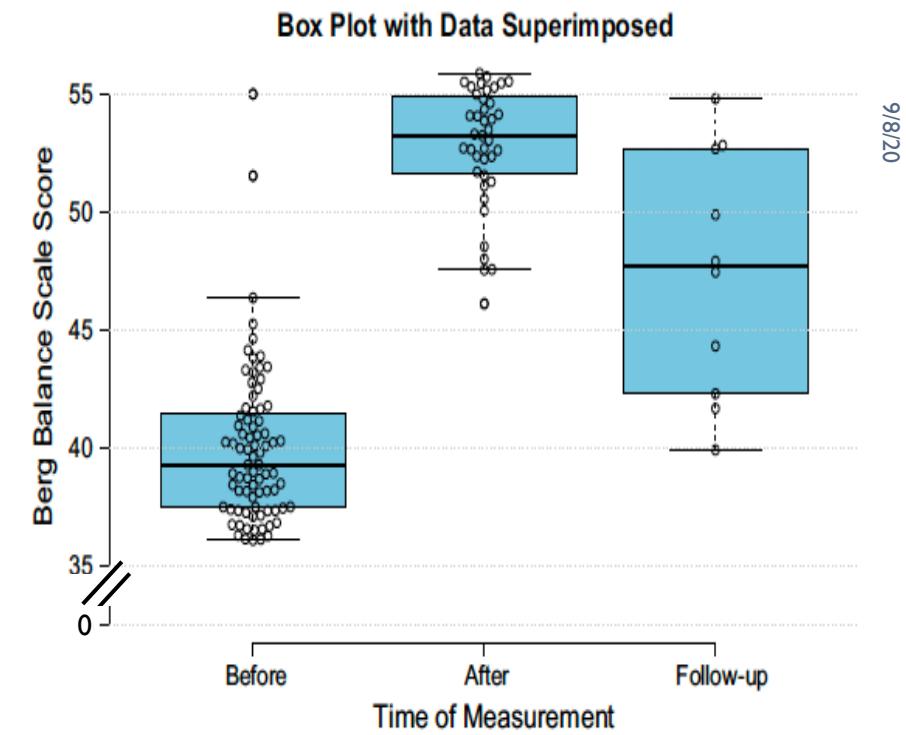


# VISUALIZATION

To avoid



Better, but ...  
Need to start from zero  
Add P-value in the figure

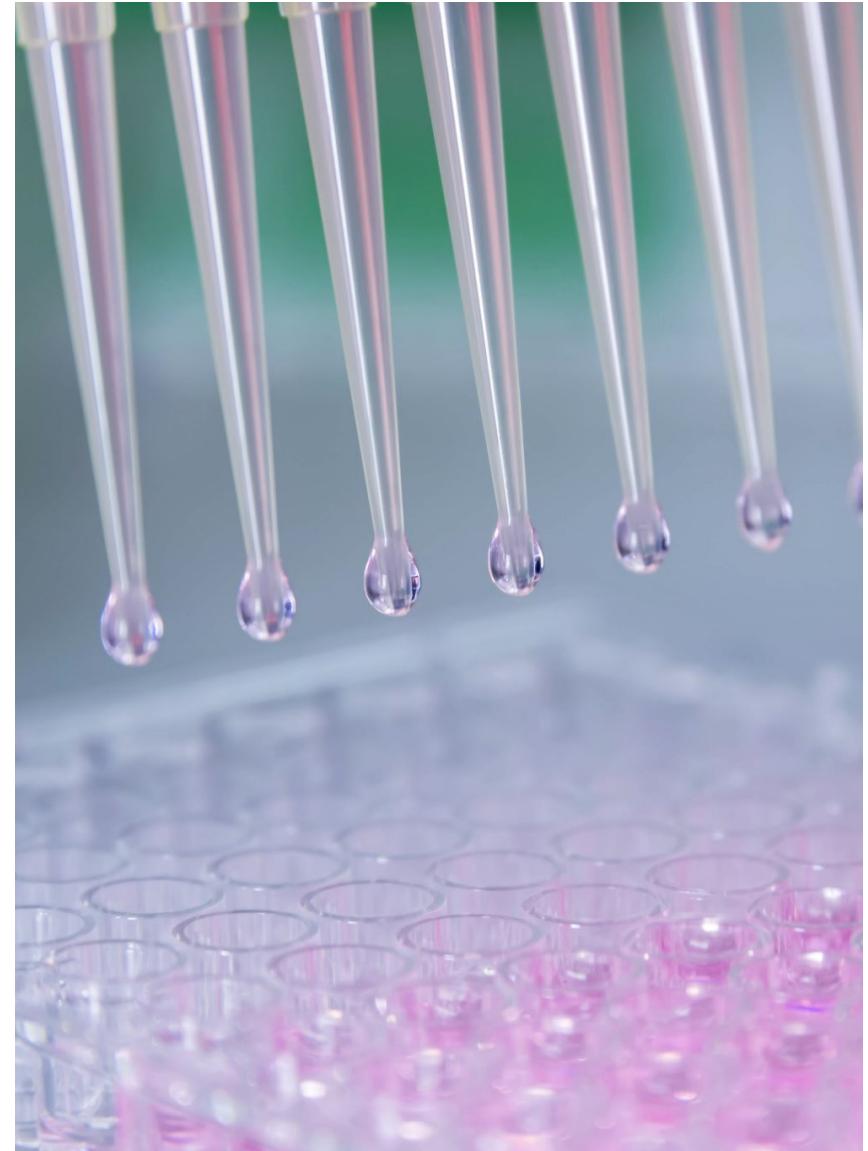


# VISUALIZATION: NOT TO FORGET

- Give information about number of replicates
  - Biological replicates and technical replicates ?

TIPHANE.MARTIN@mssm.edu

9/8/20



# BIOLOGICAL AND TECHNICAL REPLICATE?

TIPHAINEMARTIN@MSSM.EDU

9/18/20

A **technical** replicate is when you test the same sample multiple times - it's used to test the **variability in the testing protocol itself**.

A **biological** replicate is where you perform the same test on multiple samples of the same material / type of cells / tissue. The samples are different (e.g. not the same day) but are expected to be very similar (if not identical) with regard to the test. Biological replicates are used to test the **variability between samples** that were selected on the basis of being otherwise identical.

For example, if you did an RNA prep for a cell sample and placed part of that prep in multiple wells of a microtiter plate, that's a **technical replicate**.

If you then took multiple samples (eg. Different days) of the cells and made an RNA prep for each and plated them, then those are **biological replicates**. Examining the technical replicates allows you to quantify the variation internal to a test (say qPCR) and while the later would allow you to see variation in the prep procedure or sample-taking.

# VISUALIZATION: NOT TO FORGET IN THE LEGEND

- Inform about the choice of the statistical methods
  - Paired- or non-paired methods
  - One-tailed tests or two-tailed tests
  - Value of P-value and not \* . \*\* , \*\*\* or NS
  - If multiple testing, provide adjusted Pvalue using which method or nominal Pvalue  
(put information in the legend)

61.6 %: 99.19

# STATISTICAL ANALYSIS

86.72

# WHAT IS THE HYPOTHESIS NULL VS ALTERNATIVE?

## Null Hypothesis

$$H_0$$

A statement about a population parameter.

We test the likelihood of this statement being true in order to decide whether to accept or reject our alternative hypothesis.

Can include =,  $\leq$ , or  $\geq$  sign.

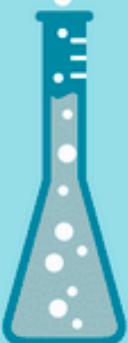
## Alternative Hypothesis

$$H_a$$

A statement that directly contradicts the null hypothesis.

We determine whether or not to accept or reject this statement based on the likelihood of the null (opposite) hypothesis being true.

Can include a  $\neq$ ,  $>$ , or  $<$  sign.



# TESTS BETWEEN TWO GROUPS

## Student's T test

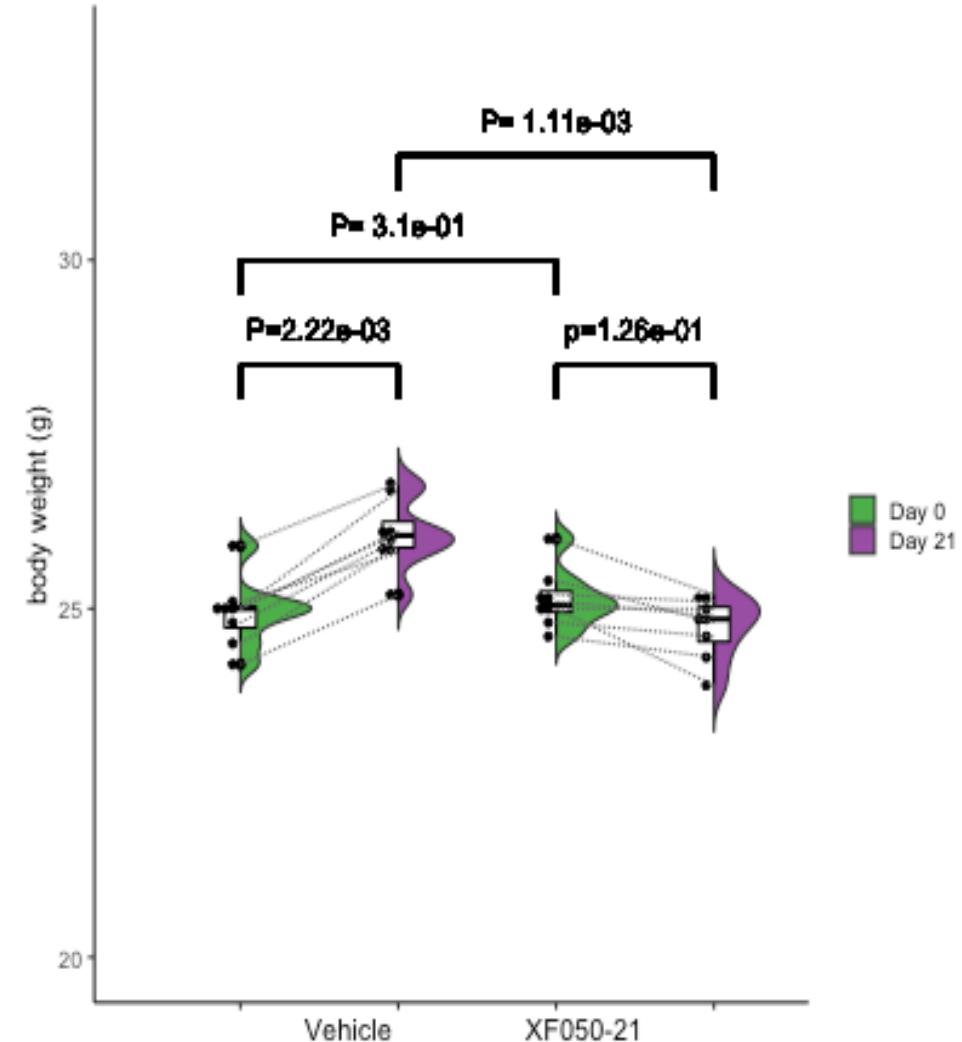
- Assumption :
  - normal distribution
  - Unrelated data
- Need to have large data
- a normal distribution
- If paired data,
  - use paired sample *t*-test

## Wilcoxon-Mann-Whitney's test

- Assumption :
  - Unrelated data
  - No assumption on the distribution of data
- Non-parametric test
- If paired data,
  - use Wilcoxon signed-rank test

# EXAMPLE OF VISUALIZATION FOR PAIRED-DATA

**Which test(s) did I use here ?**



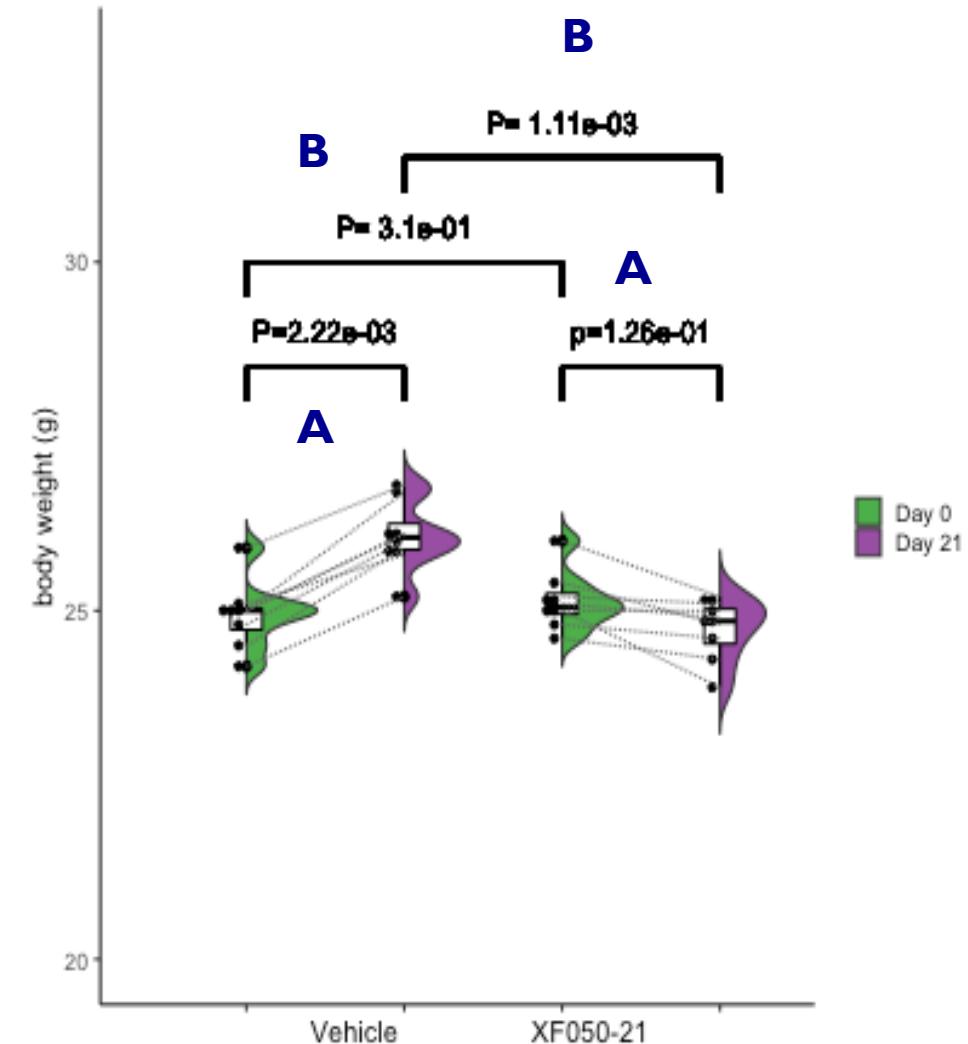
# EXAMPLE OF VISUALIZATION FOR PAIRED-DATA

**Which test(s) did I use here ?**

TIPHANE.MARTIN@MSSM.EDU

- A) Two-tailed Wilcoxon signed-rank test**
- B) Two-tailed Wilcoxon-Mann-Whitney's test**

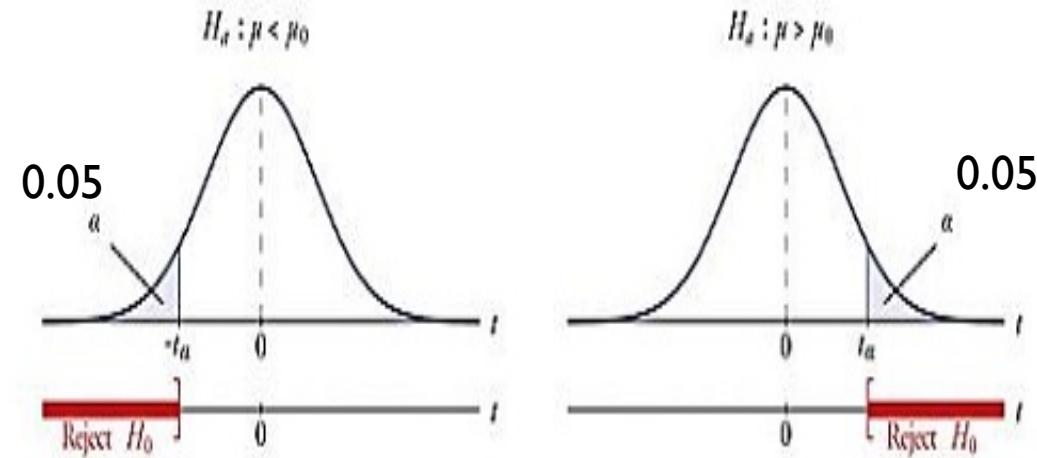
9/8/20



# ONE-TAILED TEST VERSUS TWO-TAILED TEST

- One-tailed tests:

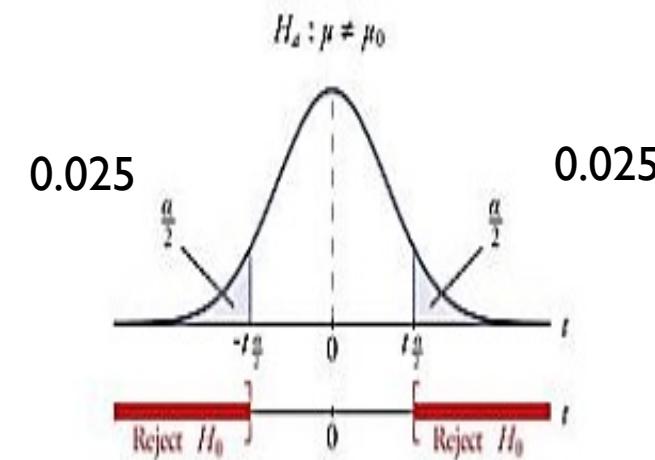
- “is the sample mean greater than  $u$  or  $P_u$ ? ”
- “is the sample mean less than  $u$  or  $P_u$ ? ”



Is the hypothesis null or alternative written here?

- Two-tailed test:

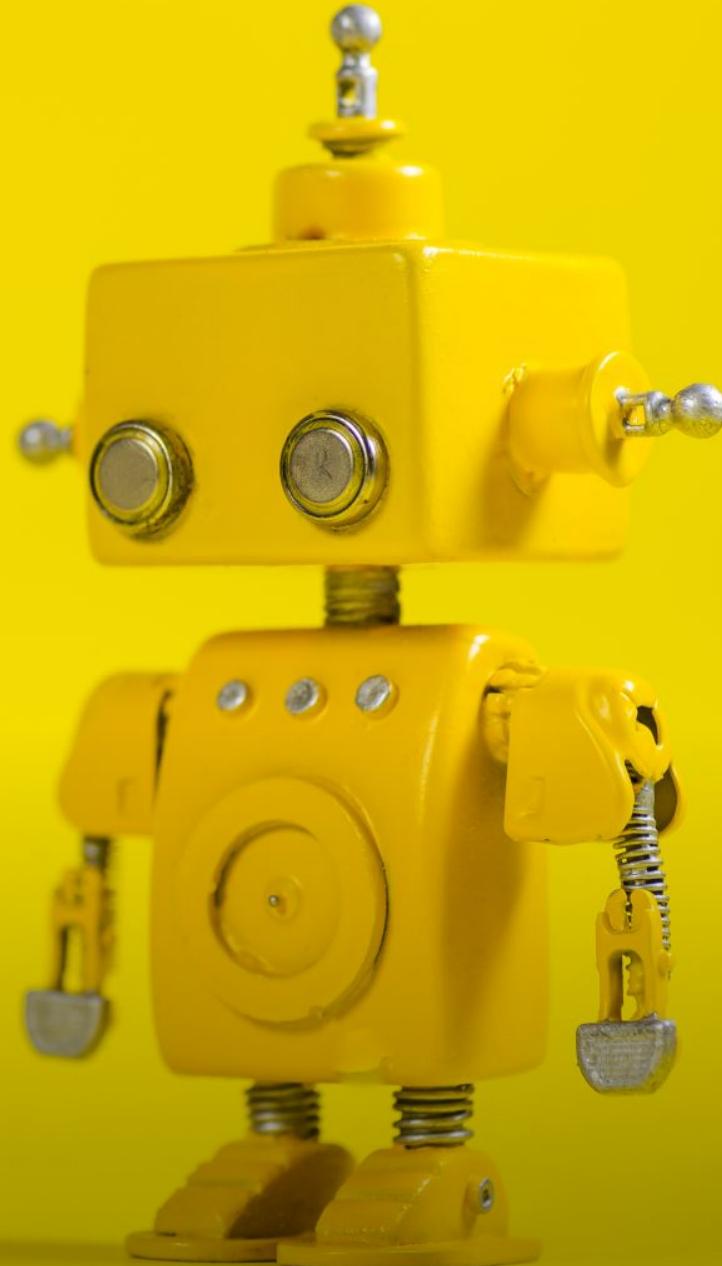
- “is there a significant difference?”



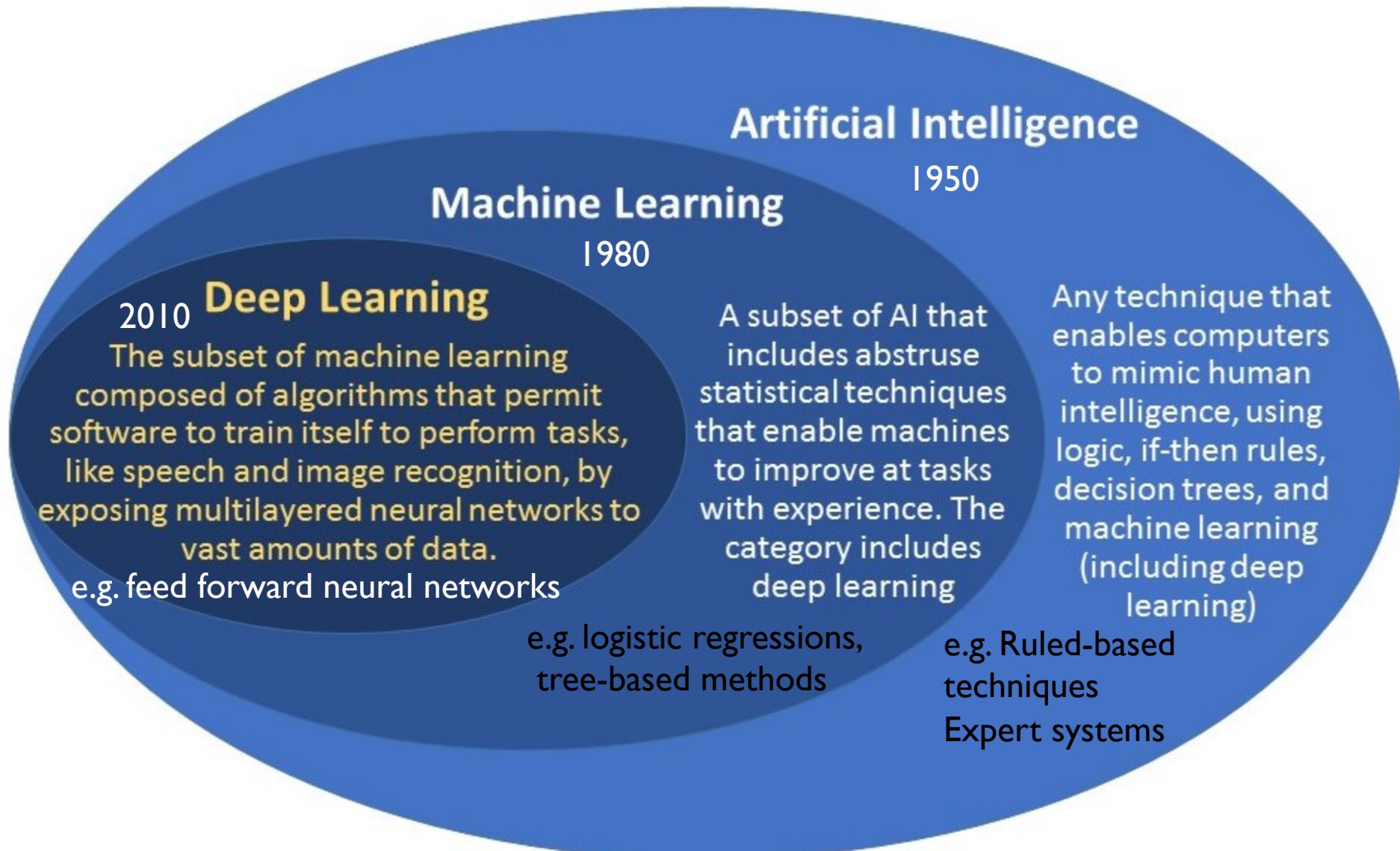
# ARTIFICIAL INTELLIGENCE (AI)

TIPHANE.MARTIN@mssm.edu

9/8/20

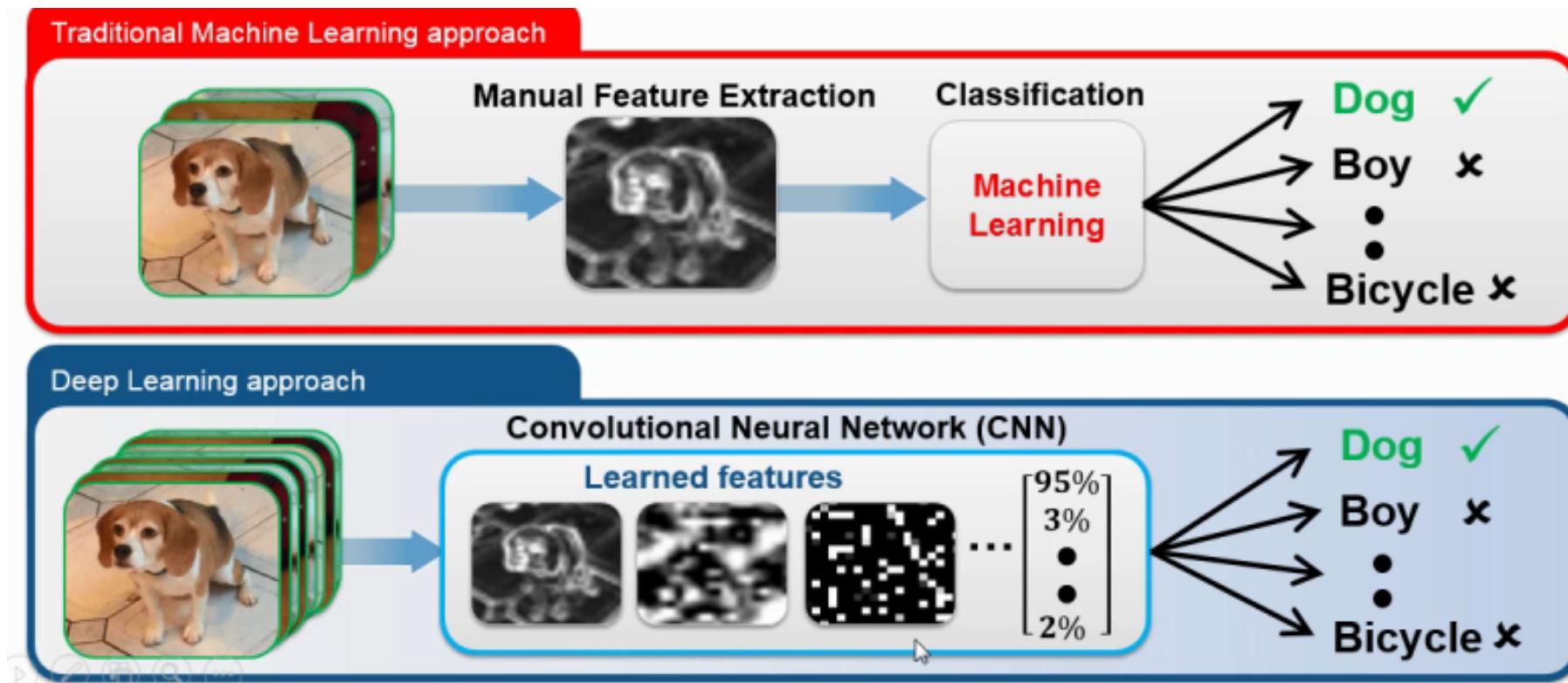


17



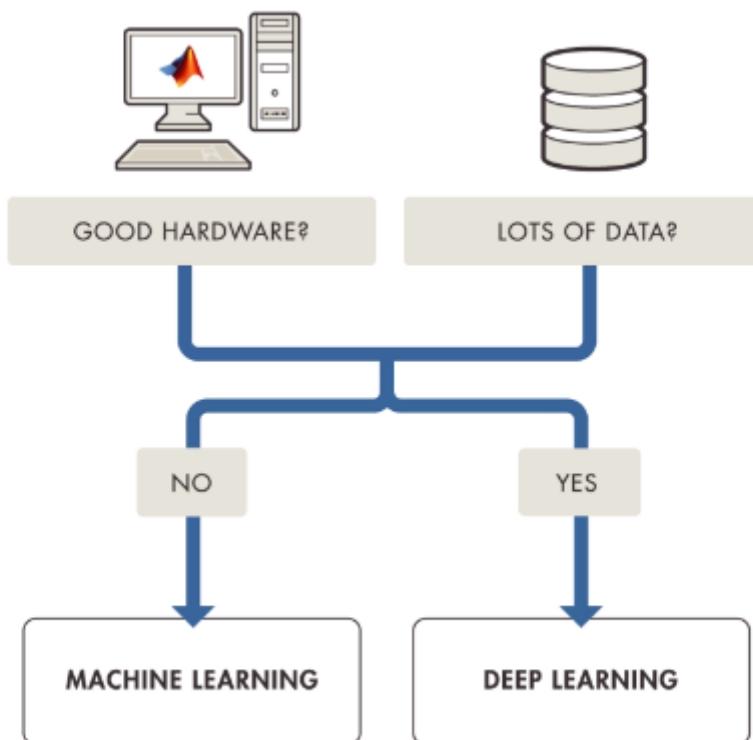
# MACHINE LEARNING (ML) VERSUS DEEP LEARNING (DL)

- Both could be used to classify (scores), predict (probabilities) or detect of features (binary)

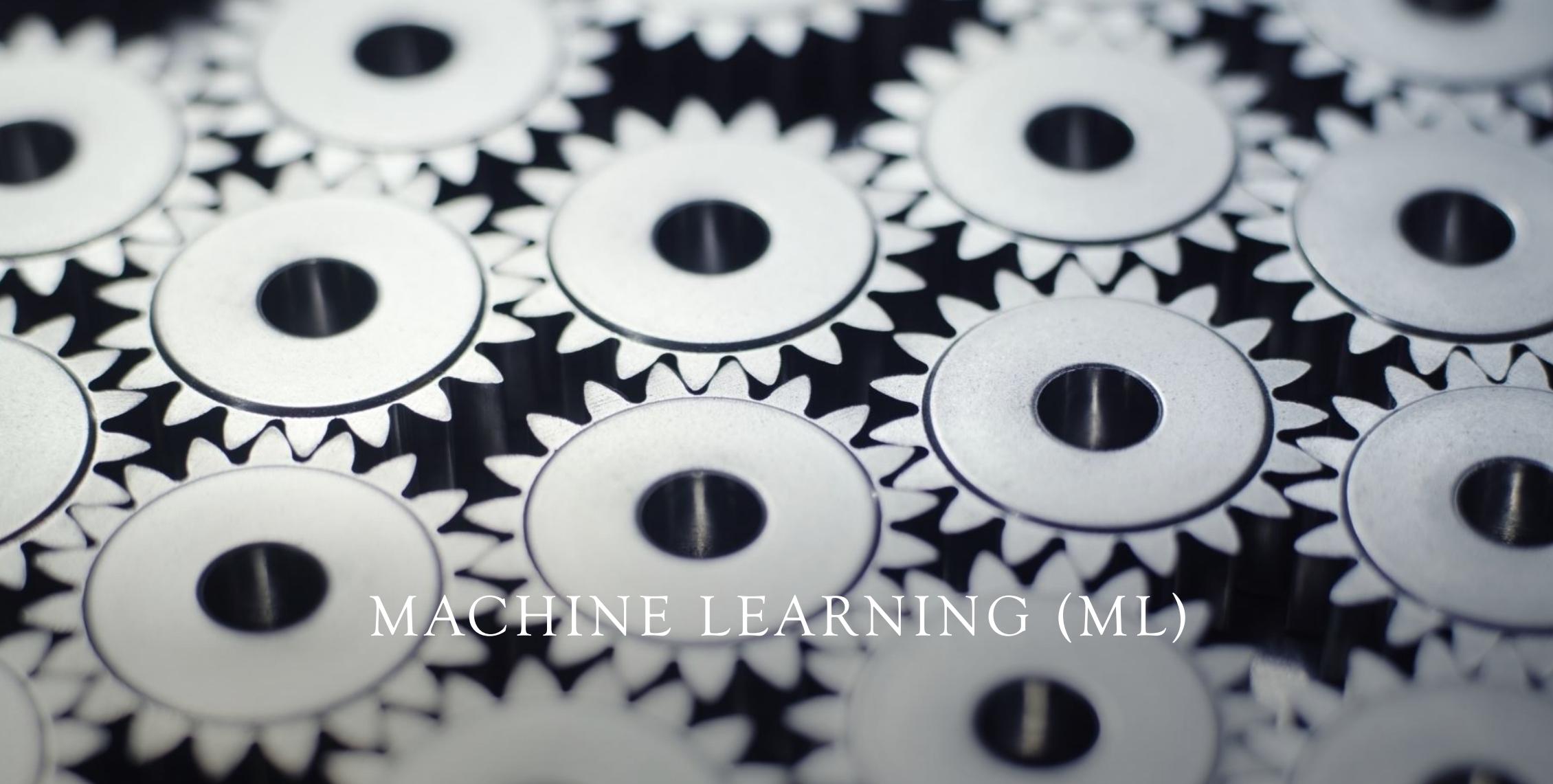


# ML VERSUS DL

	Machine Learning	Deep Learning
Training dataset	Small	Large >1000 
Choose your own features	Yes 	No
# of classifiers available	Many	Few
Training time	Short	Long GPU 



- ML : Find the good features is time consuming and can be challenging
- DL : have enough pictures to train neural networks and training is time consuming as well as the definition of layers can be challenging



# MACHINE LEARNING (ML)

# MACHINE LEARNING IN EMOJI

SUPERVISED

UNSUPERVISED

REINFORCEMENT

You are like Mr. Jourdain in “*The Bourgeois Gentleman*” (Moliere, 1670) with the prose, but instead for you, this is ML



TIPHAINE.MARTIN@MSSM.EDU

	<b>SUPERVISED</b>	human builds model based on input / output human input, machine output human utilizes if satisfactory human input, machine output human reward/punish, cycle continues
	<b>UNSUPERVISED</b>	human input, machine output human utilizes if satisfactory
	<b>REINFORCEMENT</b>	human input, machine output human reward/punish, cycle continues

## BASIC REGRESSION

	<b>LINEAR</b>	linear_model.LinearRegression() Lots of numerical data	
	<b>LOGISTIC</b>	linear_model.LogisticRegression() Target variable is categorical	or

## CLASSIFICATION

		<b>NEURAL NET</b>	neural_network.MLPClassifier() Complex relationships. Prone to overfitting Basically magic.	
		<b>K-NN</b>	neighbors.KNeighborsClassifier() Group membership based on proximity	
		<b>DECISION TREE</b>	tree.DecisionTreeClassifier() If/then/else. Non-contiguous data Can also be regression	
		<b>RANDOM FOREST</b>	ensemble.RandomForestClassifier() Find best split randomly Can also be regression	
		<b>SVM</b>	svm.SVC() / svm.LinearSVC() Maximum margin classifier. Fundamental Data Science algorithm	
		<b>NAIVE BAYES</b>	GaussianNB() MultinomialNB() BernoulliNB() Updating knowledge step by step with new info	

## CLUSTER ANALYSIS

		<b>K-MEANS</b>	cluster.KMeans() Similar datum into groups based on centroids	
		<b>ANOMALY DETECTION</b>	covariance. EllipticalEnvelope() Finding outliers through grouping	

## FEATURE REDUCTION

<b>T-DISTRIB STOCHASTIC NEIGH EMBEDDING</b>	manifold.TSNE()	Visualize high dimensional data. Convert similarity to joint probabilities	
<b>PRINCIPLE COMPONENT ANALYSIS</b>	decomposition.PCA()	Distill feature space into components that describe greatest variance	
<b>CANONICAL CORRELATION ANALYSIS</b>	decomposition.CCA()	Making sense of cross-correlation matrices	
<b>LINEAR DISCRIMINANT ANALYSIS</b>	lda.LDA()	Linear combination of features that separates classes	

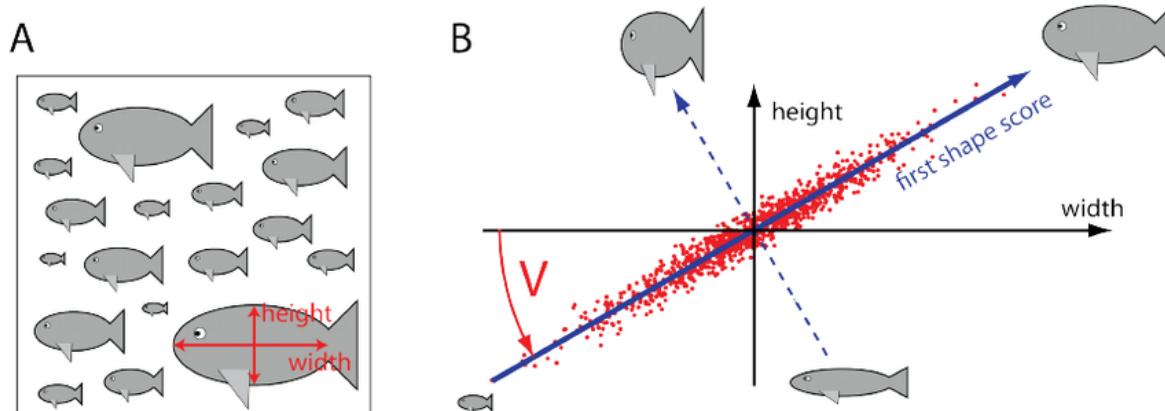
## OTHER IMPORTANT CONCEPTS

<b>BIAS VARIANCE TRADEOFF</b>	
<b>UNDERFITTING / OVERFITTING</b>	
<b>INERTIA</b>	
<b>ACCURACY FUNCTION</b>	$(TP + TN) / (P + N)$
<b>Precision Function</b>	$TP / (TP + FP)$
<b>Specificity Function</b>	$TN / (FP + TN)$
<b>Sensitivity Function</b>	$TP / (TP + FN)$

@emilyinamillion made this

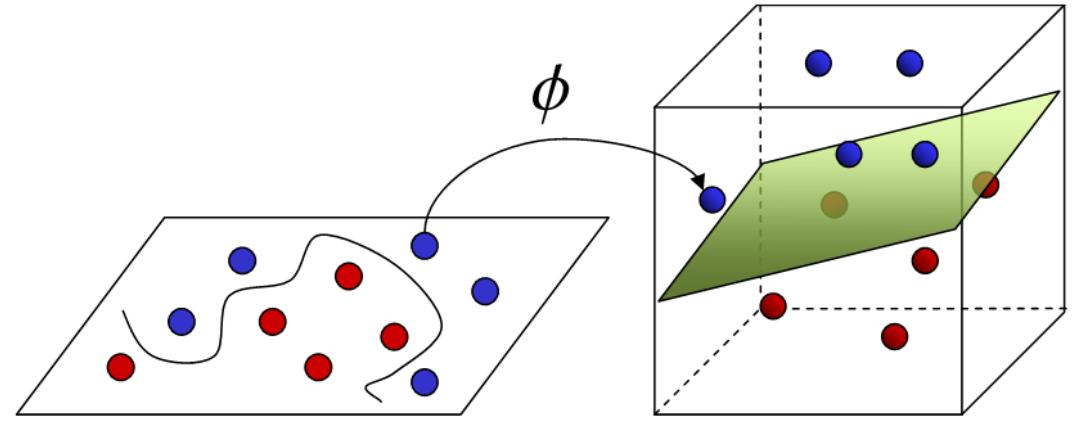
# PRINCIPAL COMPONENT ANALYSIS (PCA)

- Unsupervised
- Dimension reduction
- Orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables
- such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on



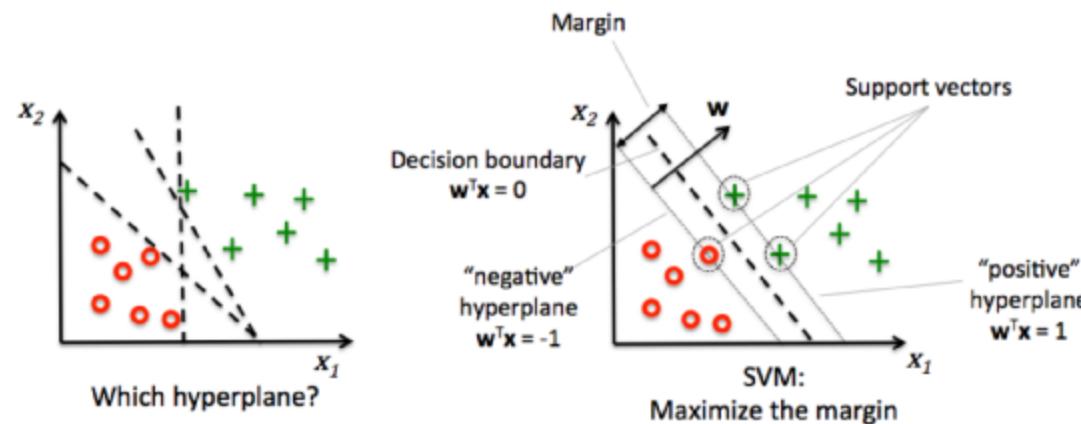
# SUPPORT VECTOR MACHINE (SVM)

- Supervised
- used for classification and regression analysis
- Model linear or not linear



Input Space

Feature Space

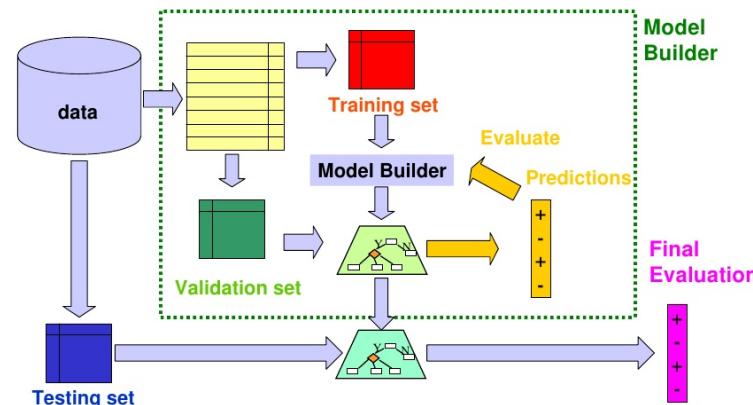
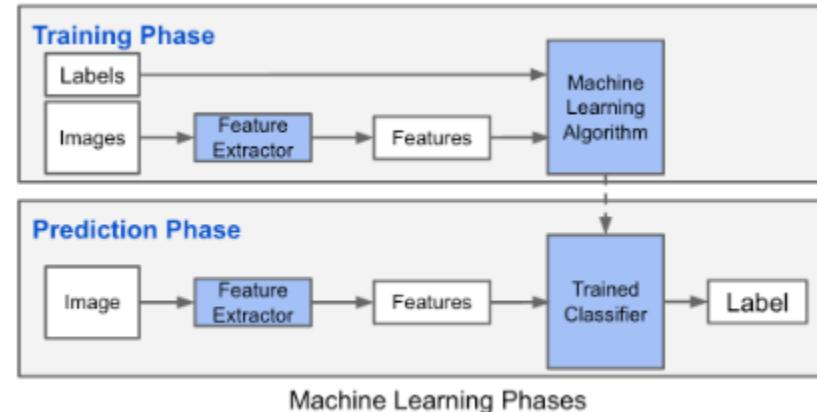


# MACHINE LEARNING TO DEEP LEARNING

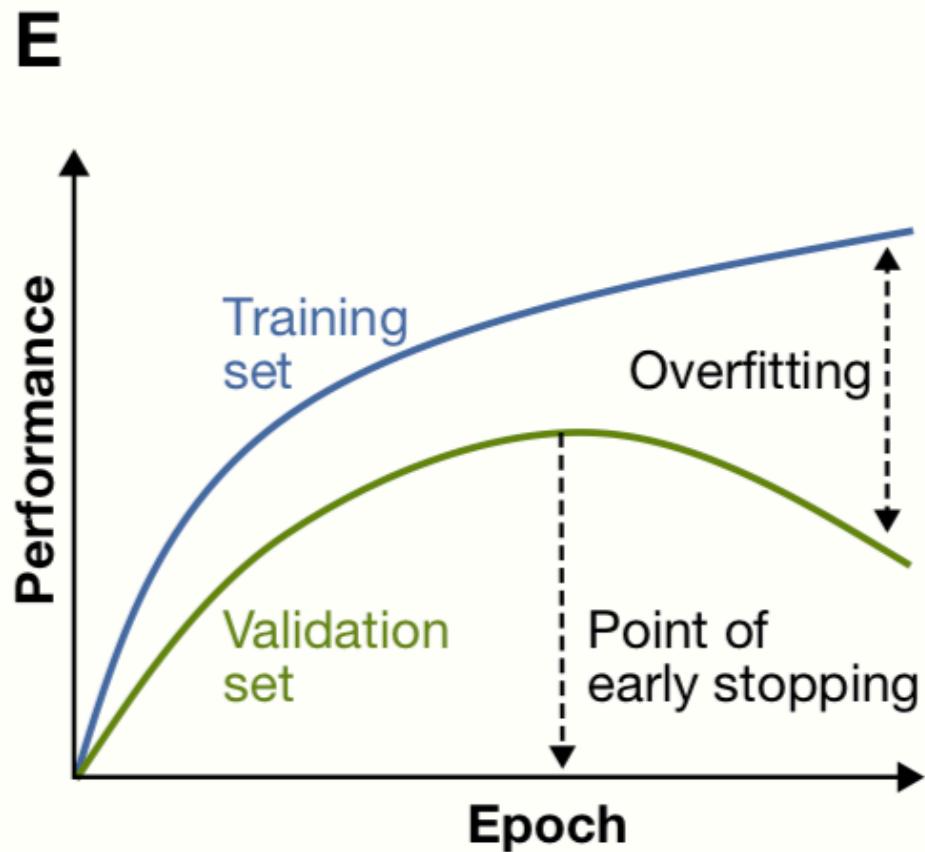
Type	Name	Description	Advantages	Disadvantages
Linear	Linear regression	The "best fit" line through all data points. Predictions are numerical.	Easy to understand – you clearly see what the biggest drivers of the model are.	<ul style="list-style-type: none"> <li>✗ Sometimes too simple to capture complex relationships between variables.</li> <li>✗ Tendency for the model to "overfit".</li> </ul>
	Logistic regression	The adaptation of linear regression to problems of classification (e.g., yes/no questions, groups, etc.)	Also easy to understand.	<ul style="list-style-type: none"> <li>✗ Sometimes too simple to capture complex relationships between variables.</li> <li>✗ Tendency for the model to "overfit".</li> </ul>
Tree-based	Decision tree	A graph that uses a branching method to match all possible outcomes of a decision.	Easy to understand and implement.	<ul style="list-style-type: none"> <li>✗ Not often used on its own for prediction because it's also often too simple and not powerful enough for complex data.</li> </ul>
	Random Forest	Takes the average of many decision trees, each of which is made with a sample of the data. Each tree is weaker than a full decision tree, but by combining them we get better overall performance.	A sort of "wisdom of the crowd". Tends to result in very high quality models. Fast to train.	<ul style="list-style-type: none"> <li>✗ Can be slow to output predictions relative to other algorithms.</li> <li>✗ Not easy to understand predictions.</li> </ul>
	Gradient Boosting	Uses even weaker decision trees, that are increasingly focused on "hard" examples.	High-performing.	<ul style="list-style-type: none"> <li>✗ A small change in the feature set or training set can create radical changes in the model.</li> <li>✗ Not easy to understand predictions.</li> </ul>
Neural networks	Neural networks	Mimics the behavior of the brain. Neural networks are interconnected neurons that pass messages to each other. Deep learning uses several layers of neural networks put one after the other.	Can handle extremely complex tasks - no other algorithm comes close in image recognition.	<ul style="list-style-type: none"> <li>✗ Very, very slow to train, because they have so many layers. Require a lot of power.</li> <li>✗ Almost impossible to understand predictions.</li> </ul>

# MACHINE LEARNING

- Classification/Prediction/Detection :
  - Unsupervised (unlabeled data)
  - Reinforced (reward)
  - Supervised (labeled data)
- Need to define and extract features
- Training/validation/test

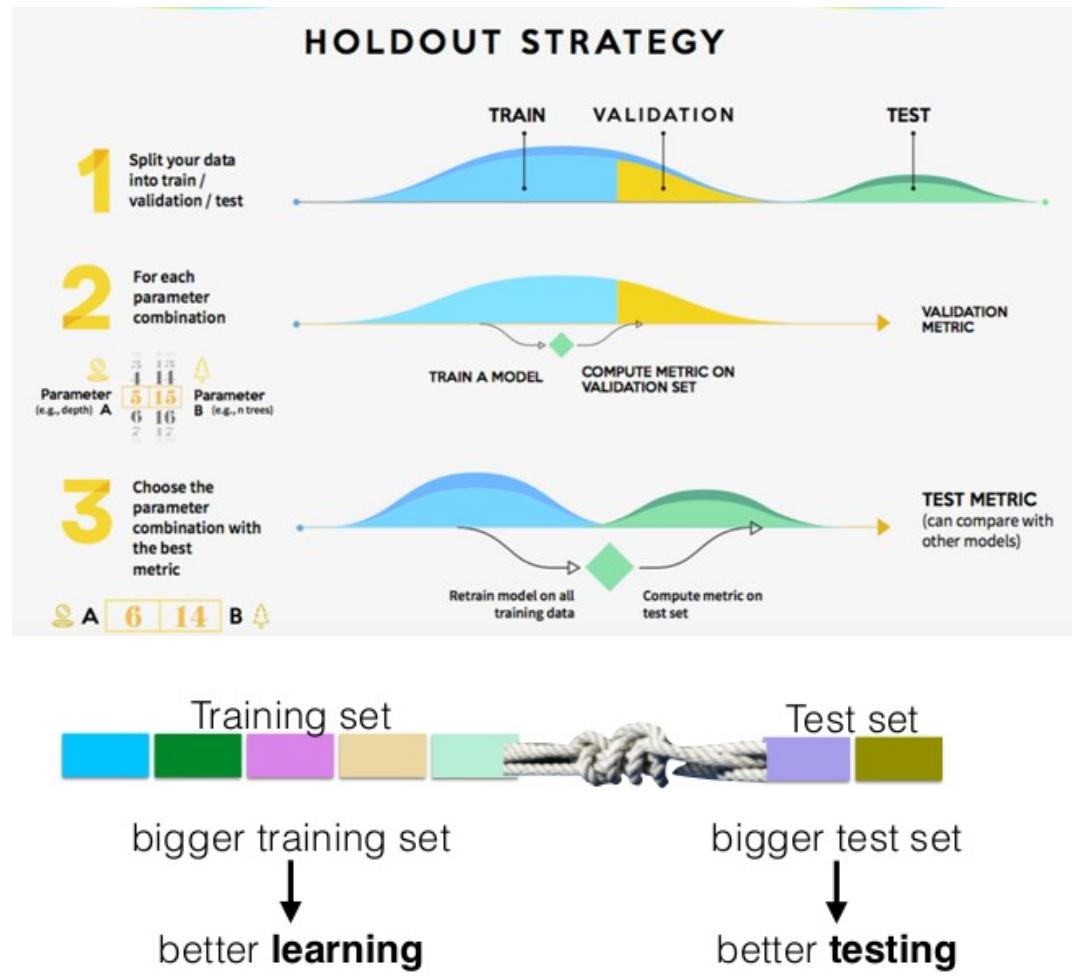


# OVERFITTING



- Results from a too complex model relative to the size of the training set
- Reducing this issue by
  - decreasing the model complexity, or
  - increasing the size of the training set

# Cross-validation (computational)



**Key:** Train & test sets must be **disjoint**.  
And the dataset or sample size is fixed.  
They grow at the expense of each other!

→ **cross**-validate  
to maximize both

