Introduction to MACHINE LEARNING

BY DATAROCKIE

- What is Data Science? 15 mins
- What is Machine Learning? 30 mins
- 4 Steps in Machine Learning 30 mins
- Algorithms You Need to Know 60 mins

What is DATA SCIENCE?

Data + Science

Andrew Ng

What is Data?

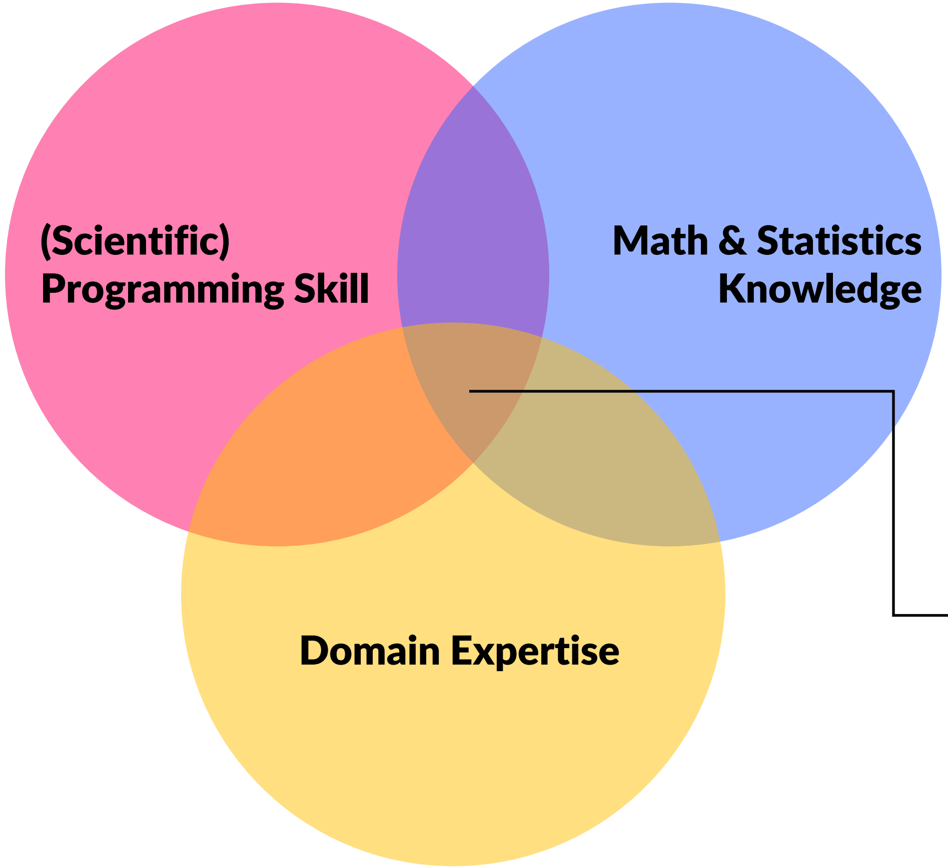
Data is new electricity.

Science as we know it.

Scientists use data to test hypothesis, theory formation.


"Data is the past, and the past is Data
Kirill Eremenko

DS Venn Diagram



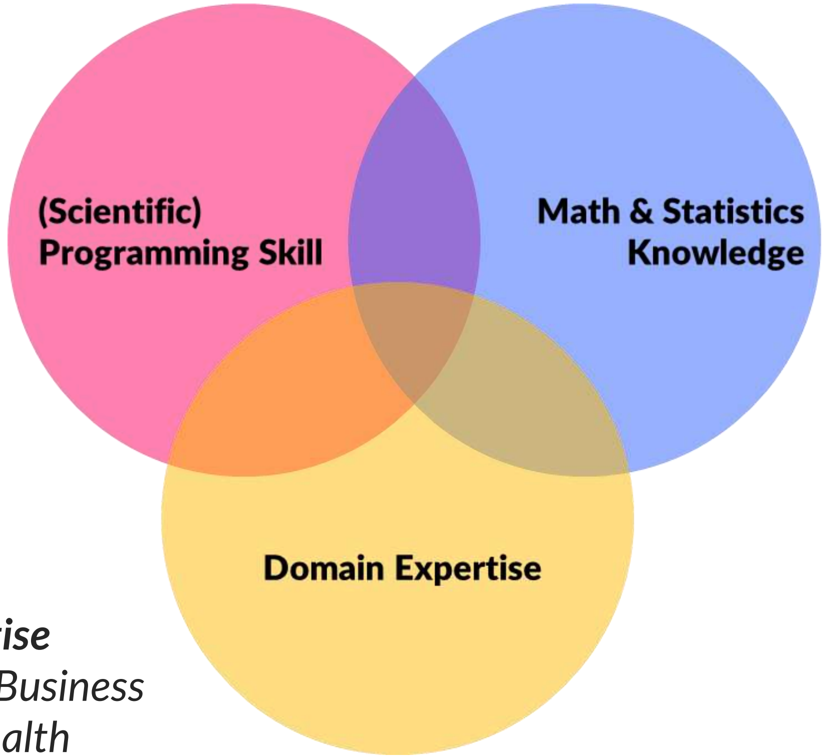
Data Science

is the intersection of three skills

http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

Essential Knowledge

Programming skill
- *R*
- *Python*
- *SQL*
- *Spark*
- *Hadoop*



Math & Statistics
- *Linear algebra*
- *Calculus*
- *Descriptive statistics*
- *Inferential statistics*
- *Bayesian statistics*
- *Probability*

Domain Expertise
- *Marketing | Business*
- *Medical | Health*
- *Politic*
- *Engineer*
- *Could be anything*

Always

Good data science starts with good question

Problem

(How to improve lives)

Data

(turn raw materials into insights to make informed decisions)

Insights

(Presentation + Deployment + Monitoring)

What is MACHINE LEARNING?

Can you guess?

x (input)	1	2	3	4	5	6	7	8	9	10
y (output)	1	4	9	16	25	36	49	64	81	

You guess the function is x^2


Ask her for a date?

Right! Can you guess this one?

	X1	X2	X3	X4	Y
Occasion 1	Restaurant	Many	Friday	Good	Yes
Occasion 2	Pub | Bar	Many	Saturday	OK	Yes
Occasion 3	Pub | Bar	Few	Monday	OK	No
Occasion 4	Restaurant	Few	Friday	Bad	Yes
Occasion 5	Restaurant	Many	Friday	Good	No
Occasion 6	Pub | Bar	Few	Saturday	OK	Yes
Occasion 7	Pub | Bar	Many	Monday	Good	No
Occasion 8	Restaurant	Few	Saturday	OK	No
Occasion 9	Pub | Bar	Many	Friday	Bad	No
Occasion 10	Restaurant	Few	Friday	Good	

Yes or No?

Machine Learning

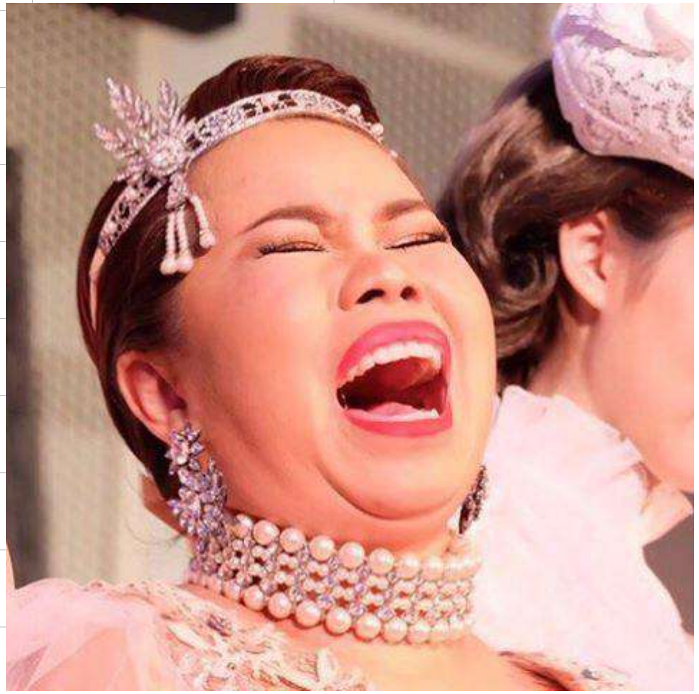
is *function approximation*

Machine Learning
is *inductive reasoning*

Learning from experience

ML algorithms trying to MAP inputs (x’s) to output (y)

	X1	X2	X3	X4	Y
Occasion 1	Restaurant	Many	Friday	Good	Yes
Occasion 2	Pub | Bar	Many	Saturday	OK	Yes
Occasion 3	Pub | Bar	Few	Monday	OK	
Occasion 4	Restaurant	Few	Friday	Bad	
Occasion 5	Restaurant	Many	Friday	Good	
Occasion 6	Pub | Bar	Few	Saturday	OK	
Occasion 7	Pub | Bar	Many	Monday	Good	
Occasion 8	Restaurant	Few	Saturday	OK	
Occasion 9	Pub | Bar	Many	Friday	Bad	
Occasion 10	Restaurant	Few	Friday	Good	



It’s good to have lots data but …

REPRESENTATION

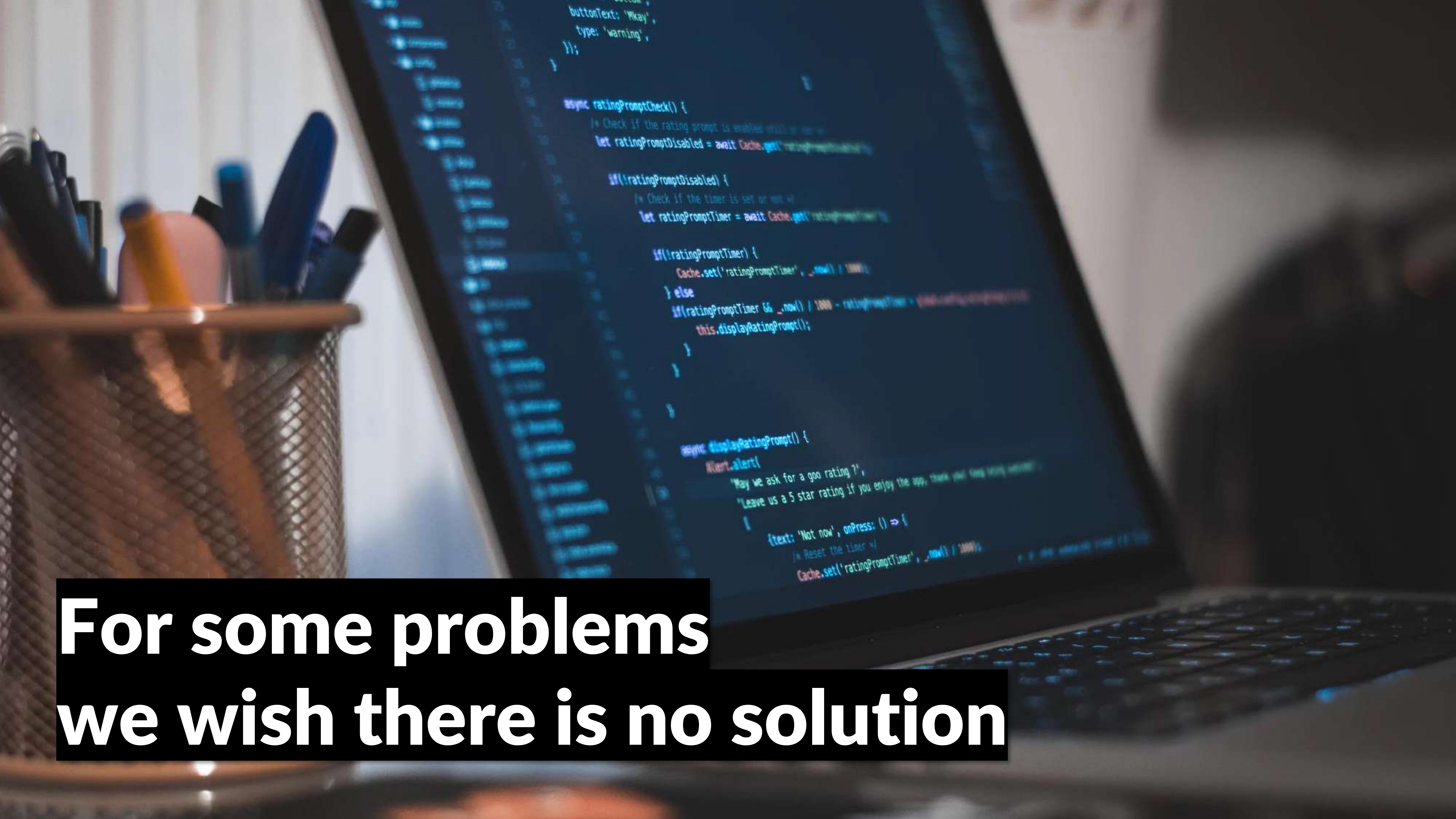
is the quality we desire

ML Example

- Predict breast cancer
- Weather forecast
- Watson analytics
- Google flu trend
- Obama winning the US election

Predicting crime before it happens


Can you get rid
of poverty?



For some problems we wish there is no solution

A proper definition of ML

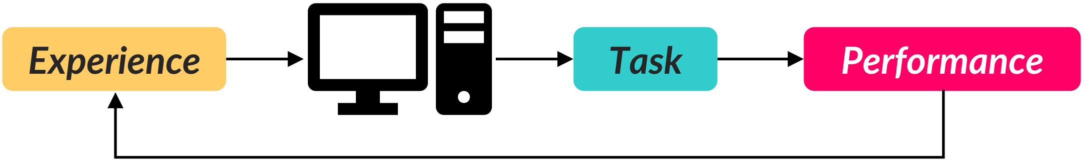
> Machine Learning: Field of study that gives computers ***the ability to learn*** without being explicitly programmed.

Arthur Samuel (1959)

A more modern def in 1998

Well posed Learning Problem: ***A computer*** *program is said to learn from* ***experience E*** *with respect to some* ***task T*** *and some* ***performance measure P****, if its performance on T, as measured by P, improves with experience E.*

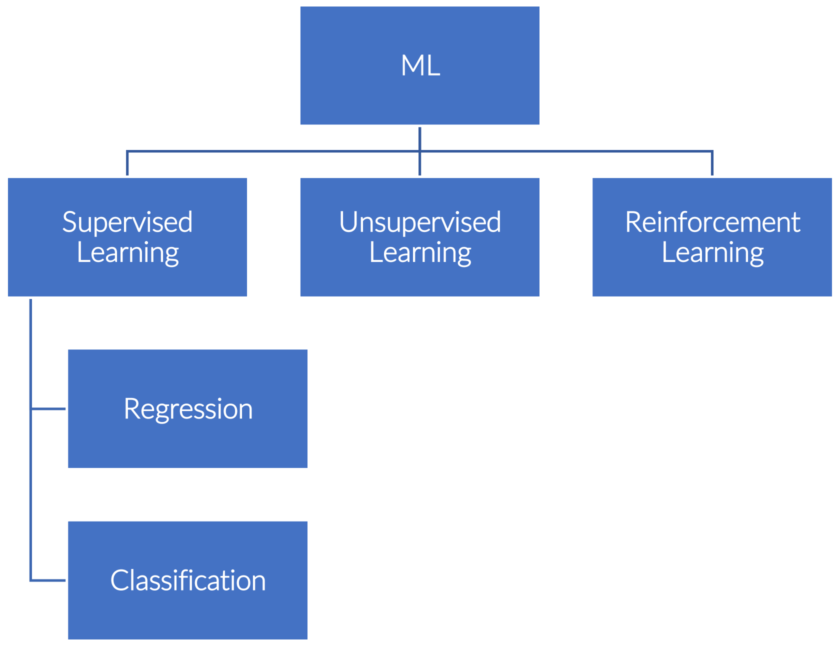
Tom Mitchell (1998)



Can we improve computer's ability to perform task T over time (without being explicitly programmed)?

Types of ML algorithms

1. Supervised Learning
    i. Regression
    ii. Classification
2. Unsupervised Learning
3. Reinforcement Learning



What’s Artificial Intelligence

Function approximation

$$Y = f(X)$$

Model = Algorithm (Data)

Predicting new unseen data

In the past

We build models to represent the world (problem we're trying to solve)

And the goal of Machine Learning is

GENERALIZATION

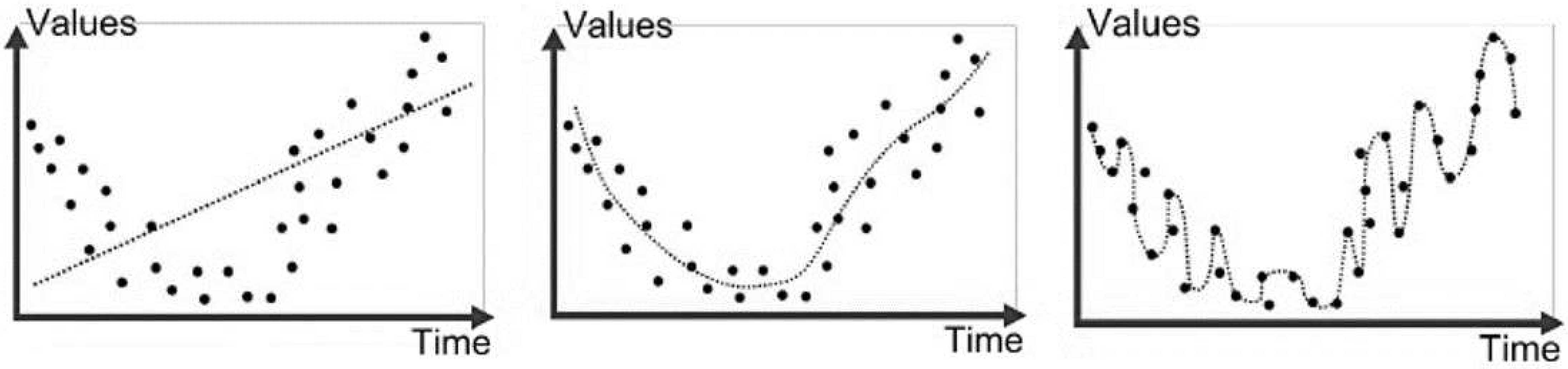
Beyond (past) data you used to build the model

Generalization means we make sure our models **don't overfit** the data

Also don't underfit

We mean data in the past

If your model ***can't generalize to new unseen data***, then you're in trouble.



Underfitting ***Good fit*** ***Overfitting***

https://cdn-images-1.medium.com/max/1600/1*6vPGzBNppqMHllg1o_se8Q.png

This model looks okay as
it fit underlying patterns
in the data well. Though,
not 100% correct ?

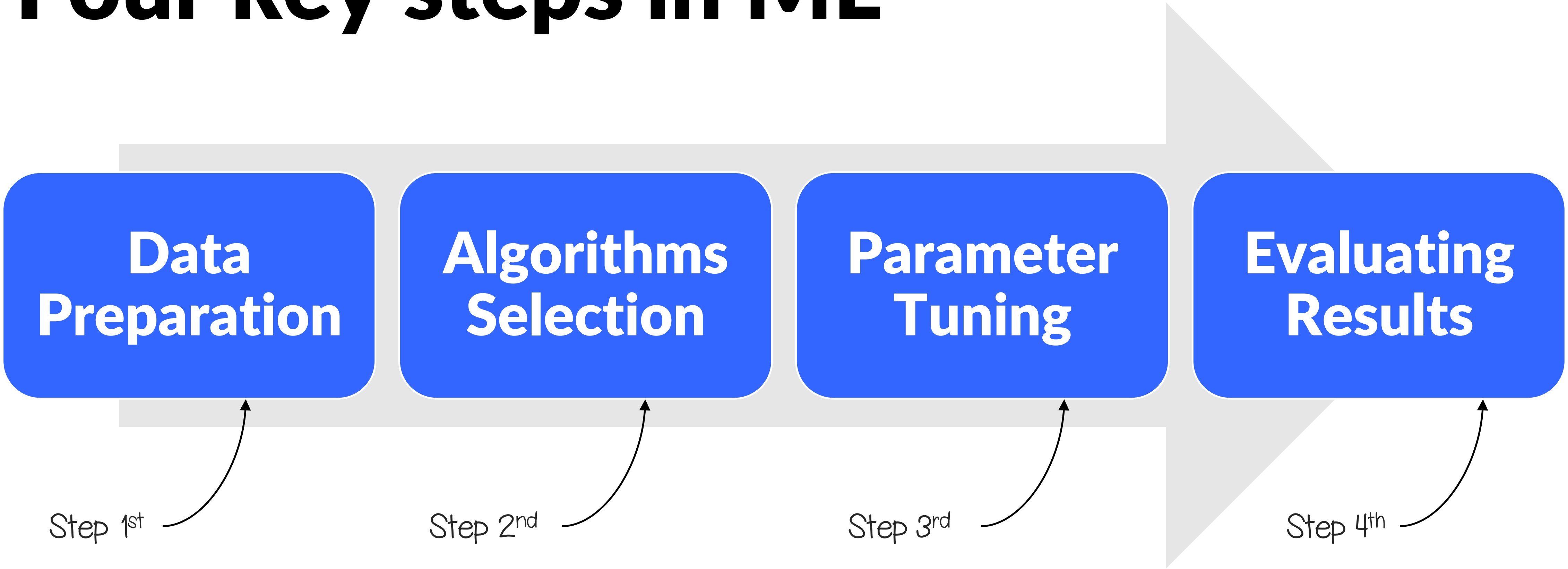
SUMMARY

- Data Science in Thailand is still very young
- ML is data scientist's arsenal tool
- ML is **Function Approximation**, learning from experience
- **Representation** & **Generalization** (avoid overfitting | underfitting)
- 3 types of ML algorithms including supervised, unsupervised and reinforcement learning
- Regression predicts numbers | Classification predicts categories

KICK START

MACHINE LEARNING

Four key steps in ML



DATA PREPARATION

Data Format

Variables | Features

Target Variable

	X1	X2	X3	X4	Y
Occasion 1	Restaurant	Many	Friday	Good	Yes
Occasion 2	Pub | Bar	Many	Saturday	OK	Yes
Occasion 3	Pub | Bar	Few	Monday	OK	No
Occasion 4	Restaurant	Few	Friday	Bad	Yes
Occasion 5	Restaurant	Many	Friday	Good	No
Occasion 6	Pub | Bar	Few	Saturday	OK	Yes
Occasion 7	Pub | Bar	Many	Monday	Good	No
Occasion 8	Restaurant	Few	Saturday	OK	No
Occasion 9	Pub | Bar	Many	Friday	Bad	No
Occasion 10	Restaurant	Few	Friday	Good	No

Data Points | Observations

Variable Types

1. Binary (1/0)
2. Categorical
3. Integer
4. Continuous

Feature Selection

- Choose relevant set of features that represent the problem
- Curse of dimensionality: 2^n

Feature Engineering

- Recode
- Combine multiple variables (PCA)
- Extract new information from existing feature

Gentle intro to Feature Engineering

ID	Name	DOB	Math	Stats	Art	Music	Design
1	Mr. David Beckham	13 / 05 / 1975	85	90	50	42	49
2	Ms. Angelina Jolie	20 / 09 / 1988	50	60	90	95	88
3	Mr. Khalid Khan	05 / 01 / 1950	32	65	70	72	65
4	Mr. Hibino Takamoto	31 / 12 / 1995	75	76	80	50	30

Question:

What new variable can we create from this dataset?
Think at least 5 new variables

We can use
dplyr::mutate( ) to
create new variables

Missing Data

Mean | Median imputation

1. Approximated
2. Predicted (Supervised Learning)
3. Removed

If we have large data points and NA is not much, we may consider removing them

Approximated or Removed?

Console ~/Desktop/R PROGRAMMING/

```
> head(complete.cases(biopsy), 200)
  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [14]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
 [27]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [40]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [53]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [66]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [79]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [92]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[105]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[118]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[131]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[144]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[157]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
[170]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[183]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[196]  TRUE  TRUE  TRUE  TRUE  TRUE
```

R functions:

complete.cases(df) สำหรับตรวจสอบว่าข้อมูลมี NA หรือเปล่า?

tidyr::drop_na(df) สำหรับ drop case ที่มีข้อมูลไม่ครบทิ้ง

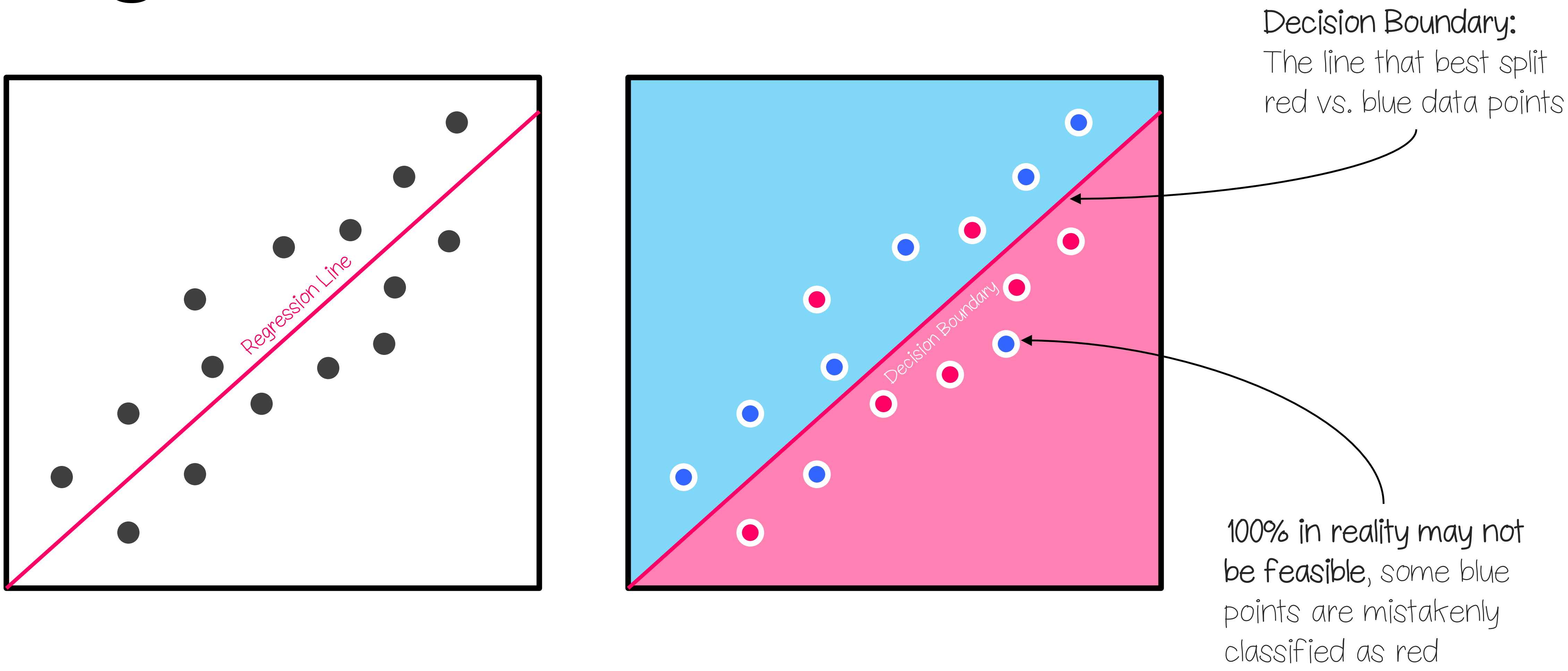
ALGORITHM SELECTION

There are 3 types of algorithms

	Algorithms	**Main Tasks**
Unsupervised learning	*K-Means* *Principal Component Analysis (PCA)* *Association Rules* *Social Network Analysis (SNA)*	*Tell me what patterns exist in my data* ***- descriptive***
Supervised learning	*Linear Regression* *Logistic Regression* *Decision Tree* *Random Forests* *KNN* *Support Vector Machine* *Neural Networks*	*Use the pattern in my data to make predictions* ***- predictive***
Reinforcement learning	*Multi-Armed Bandits (UBC, Thompson Sampling)*	*Use the pattern in my data to make predictions and* ***– improve –*** *these predictions as more results come in*

Source: Data Science For The Layman (2017)

Regression vs. Classification



NO FREE LUNCH

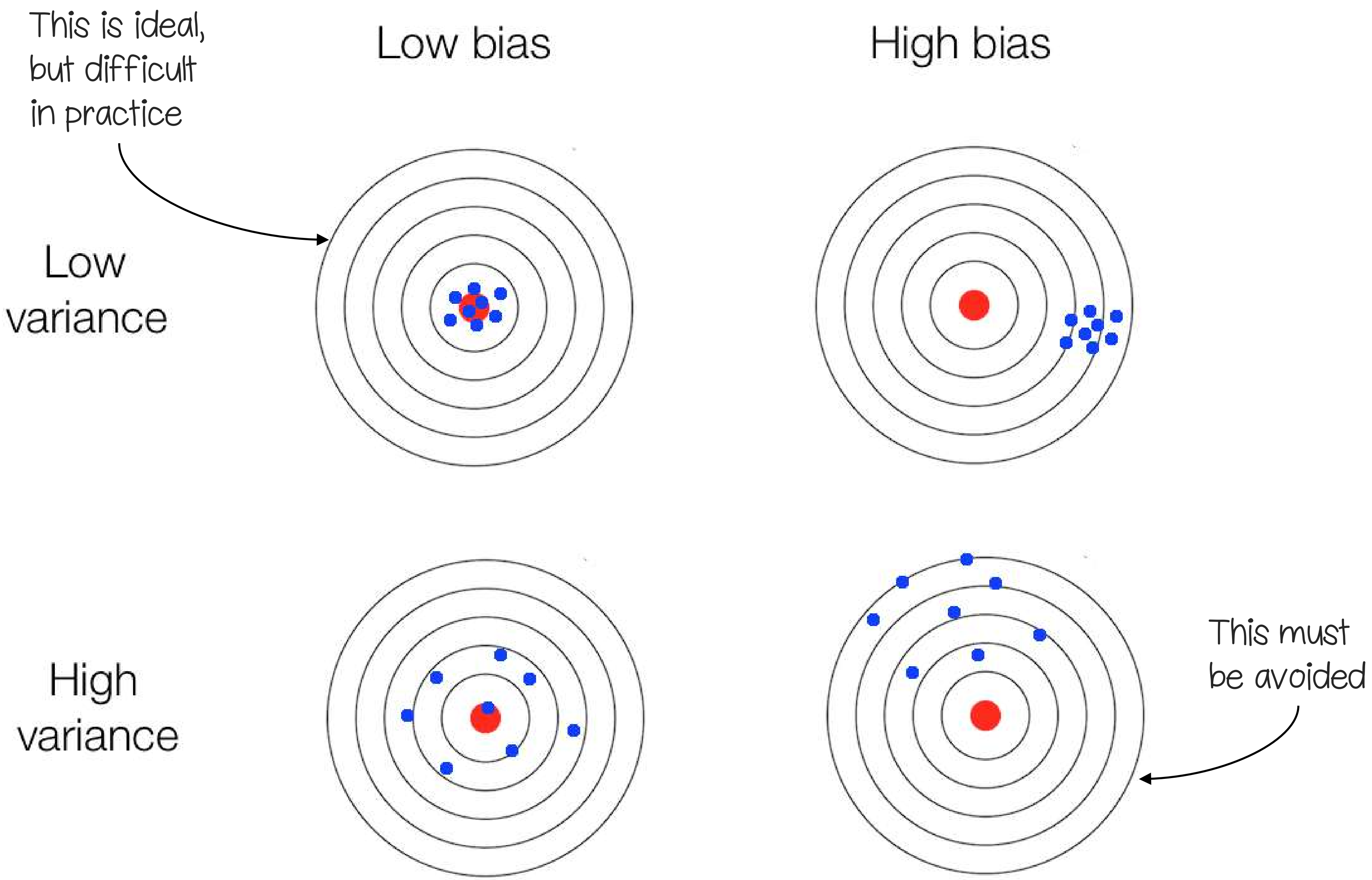
- Speed
- Accuracy
- Bias & Variance Tradeoff

Our model consists of 3 types of error

$$Y = f(X) + \text{Error}$$

- Bias
- Variance
- Irreducible error

Trade off



https://towardsdatascience.com/data-science-simplified-key-concepts-of-statistical-learning-45648049709e

Example Models

High Bias *Having lots of assumptions to do function approximation*	*Linear Regression* *Logistic Regression* *Linear Discriminant Analysis*	***Low Variance*** *New data does not cause model to change much*
Low Bias *Having little or few assumptions about the function approximation*	*Decision Tree* *KNN* *SVM*	***High Variance*** *New data causes model to change much i.e. prediction results vary significantly compared to low variance algorithms*

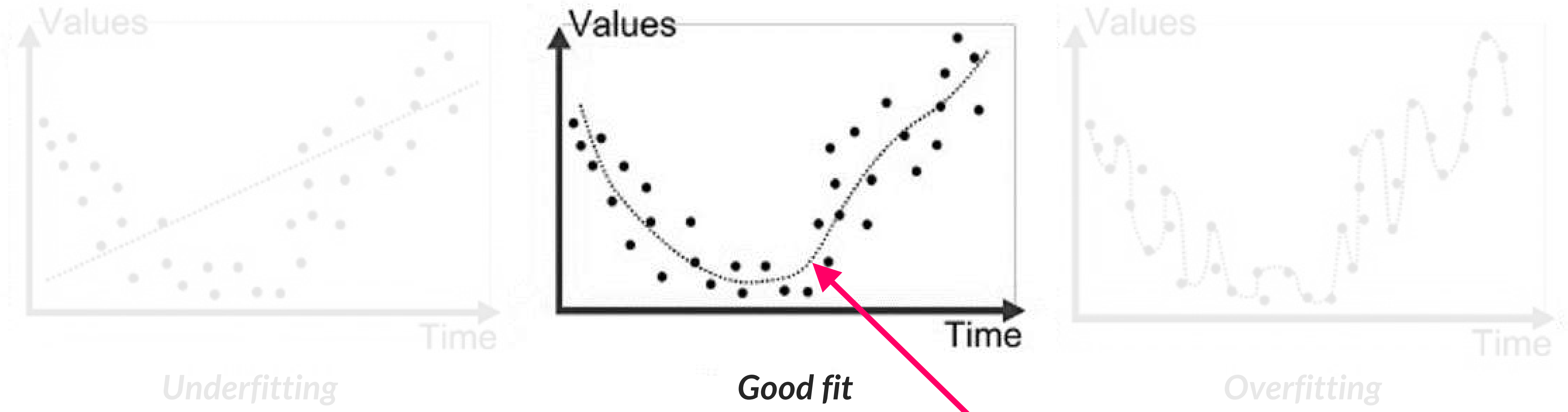
PARAMETER TUNING

Music Composer is a very good example

There are so many learners.

Even the <u>same</u> learner can have <u>different</u> performances.

We tune parameter to get the best results (GOOD FIT) from our learner.



https://cdn-images-1.medium.com/max/1600/1*6vPGzBNppqMHllg1o_se8Q.png

*The parameters that can be tuned called "**HYPERPARAMETERS**"*

Some models have hyperparameters, some don't.

Stochastic Gradient Boosting

```
method = 'gbm'
```

Type: Regression, Classification

Tuning parameters:

- `n.trees` (# Boosting Iterations)
- `interaction.depth` (Max Tree Depth)
- `shrinkage` (Shrinkage)
- `n.minobsinnode` (Min. Terminal Node Size)

Hyperparameters of GBM algorithms

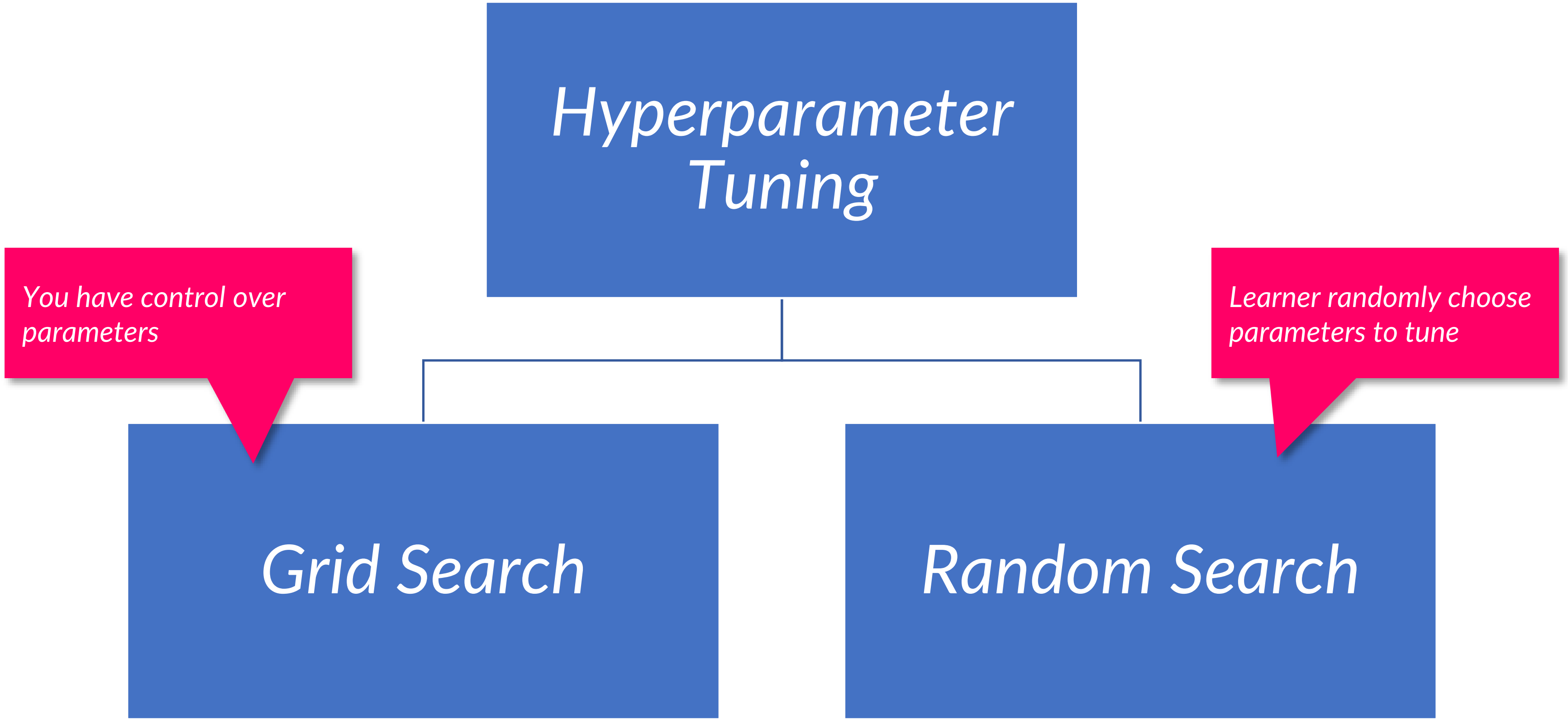
Required packages: `gbm` , `plyr`

A model-specific variable importance metric is available.

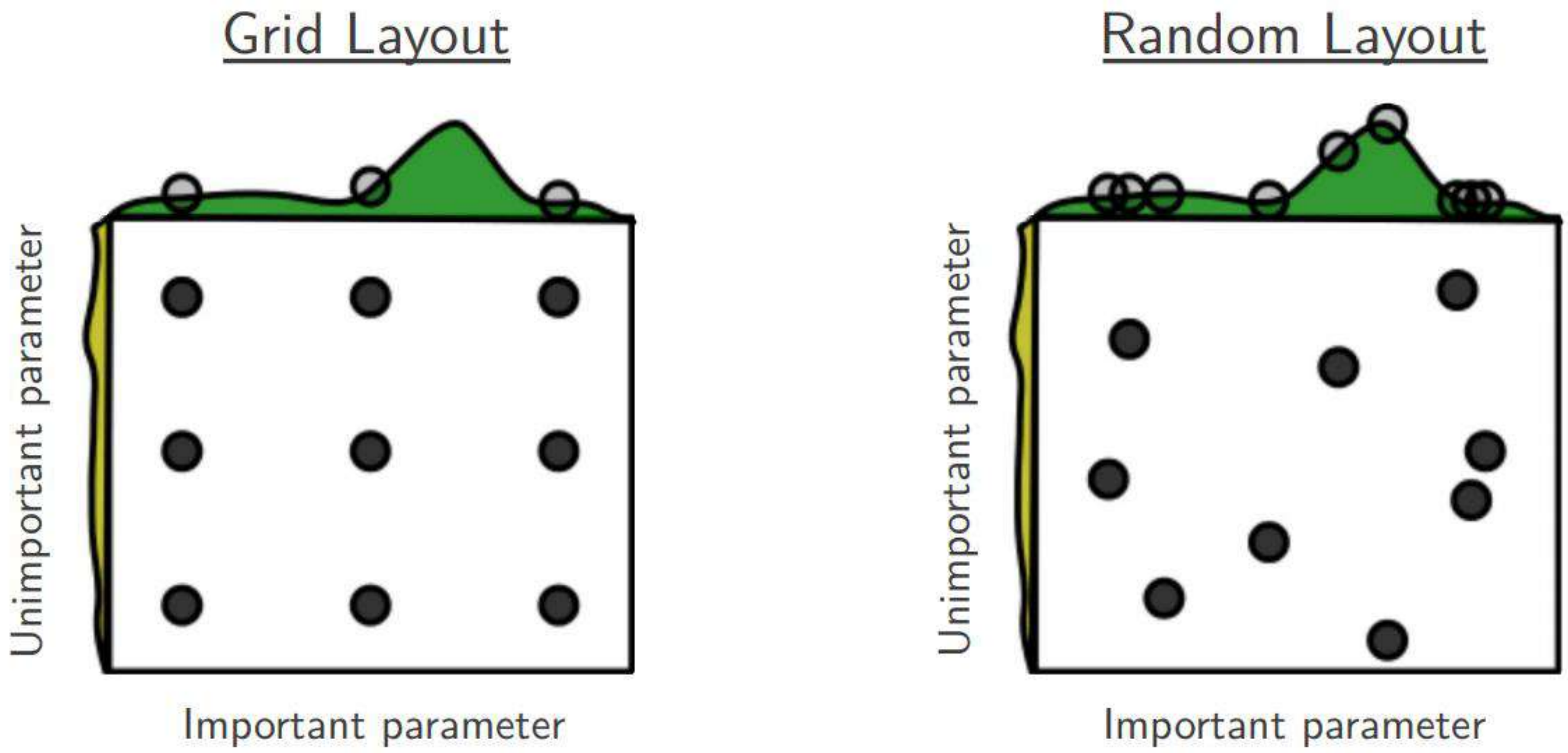
http://topepo.github.io/caret/train-models-by-tag.html

SEARCH for the best tune

ML is optimization problem



It really depends on the learners. Sometimes it's more efficient to use random search, sometimes it's not.



http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

EVALUATING RESULTS

Performance Metrics

Classification problems

- *Confusion Matrix*
- *% Accuracy*
- *Area Under Curve (AUC)*
- *Logarithmic Loss (Log Loss)*
- *Cohen's KAPPA #for imbalanced problems*

Regression problems

- *Mean Absolute Error (MAE)*
- *Root Mean Squared Error (RMSE)*

Cancer Detection

Confusion Matrix

Total n = 100	Reference	
Predicted	**Cancer**	**No Cancer**
Cancer	60	3
No Cancer	2	35

Accuracy is the sum of diagonal divided by total sample size

60+35 / 100 = **95%**

Confusion Matrix

Total n = 100	**Reference**	
Predicted	Cancer	No Cancer
Cancer	60 *(A)*	3 *(B)*
No Cancer	2 *(C)*	35 *(D)*

ถ้าผู้ป่วยเป็นมะเร็ง เราทายถูกเท่าไร? (%)

***Sensitivity** = A / A+C*

ถ้าผู้ป่วยไม่ได้เป็นมะเร็ง เราทายถูกเท่าไร? (%)

***Specificity** = D / B+D*

ผล prediction "cancer" ของเราถูกกี่ %

***Precision** = A / A+B*

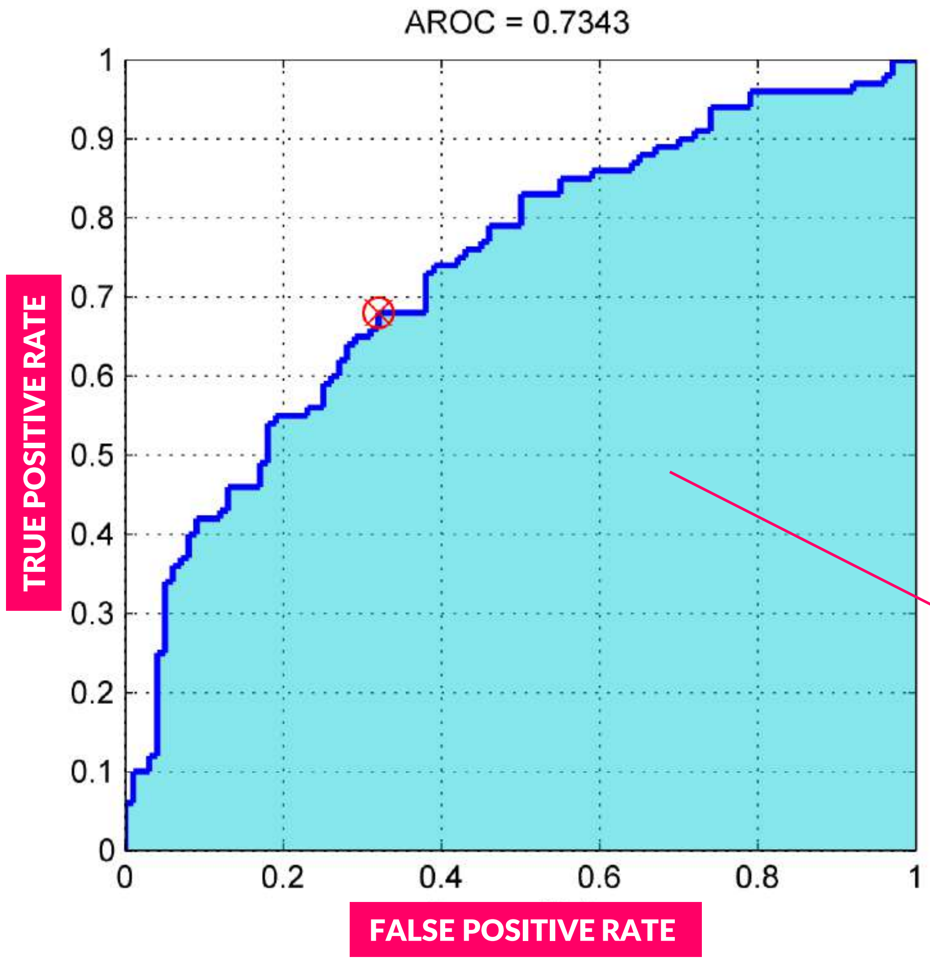
ในผู้ป่วยโรคมะเร็งทั้งหมด เราทำนายถูกเท่าไร? (%)

***Recall** = A / A+C*

Measure of overall test's accuracy

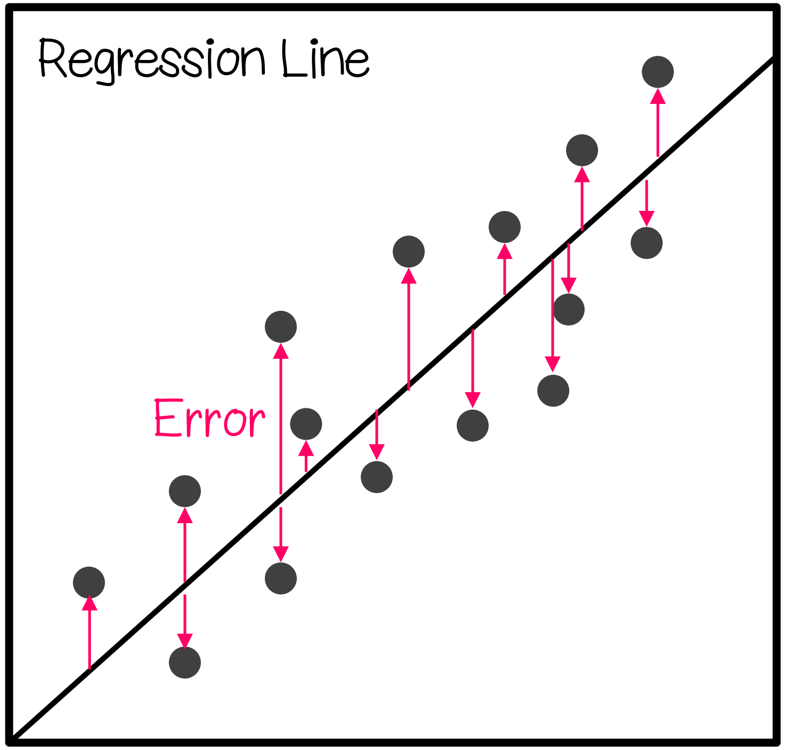
***F1** = 2 * (Precision * Recall) / (Precision + Recall)*

Area Under Curve (AUC)



For binary classification, sometimes we want to maximize AUC

Root Mean Squared Error (RMSE)



Mathematical Expression always look scary

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Actual - Predicted\ y_i)^2}{N}}$$

$$RMSE = \sqrt{\frac{sum(error^2)}{N}}$$

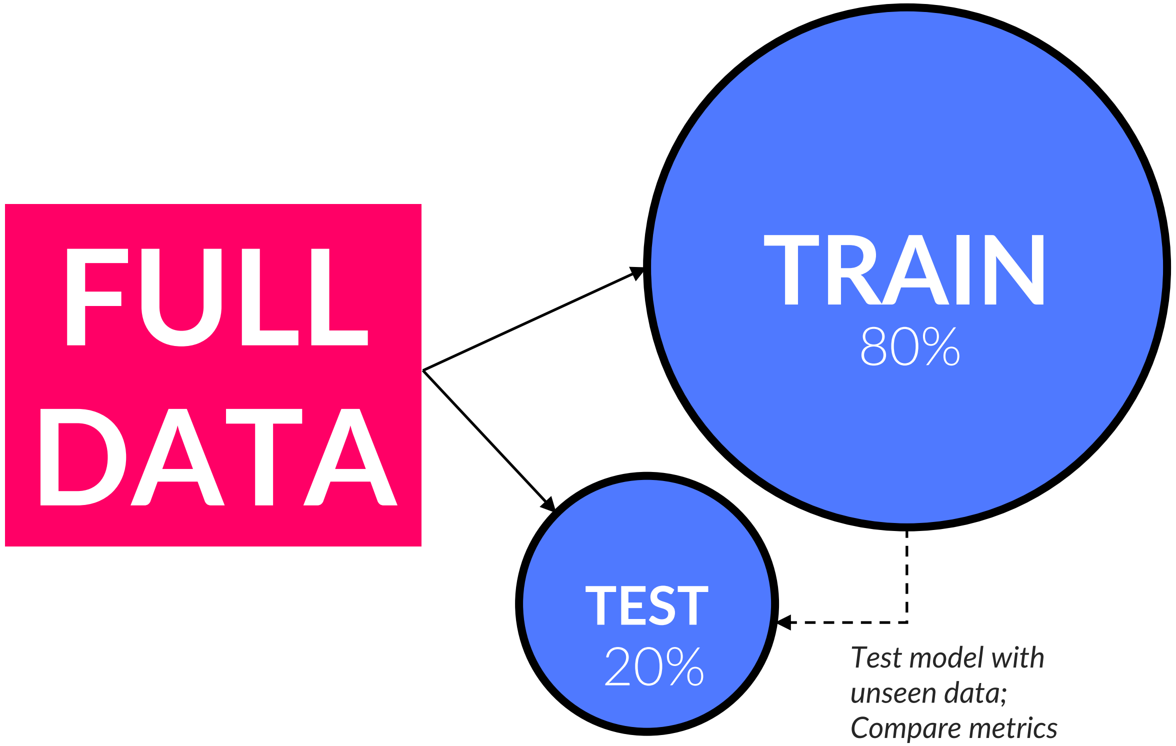
Remember this one

$$RMSE = \sqrt{mean\ (error^2)}$$

Recall the goal of ML is "GENERALIZATION"

We must test our model on new unseen dataset

Is our model good to go?



Training set (build model)

Testing set (evaluate model)

Decision Criterion:

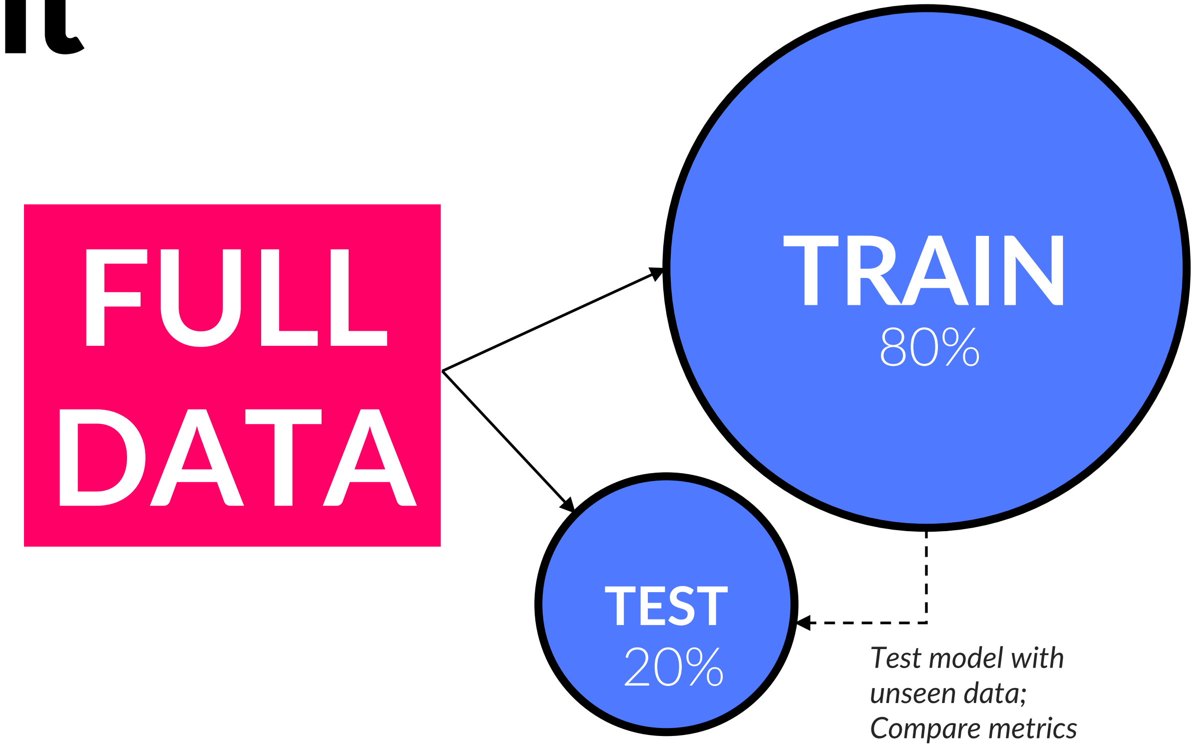
*The model **passes the test** if tested performance is similar trained performance [metric]*

Resampling techniques

1. Train-test split
2. *k-Fold Cross-Validation*
3. *Repeated k-Fold Cross-Validation*
4. *Bootstrap*
5. *Leave One Out Cross-Validation*

This method is preferred but it's very expensive for large dataset

The most efficient method is train-test split



K-Fold Cross Validation

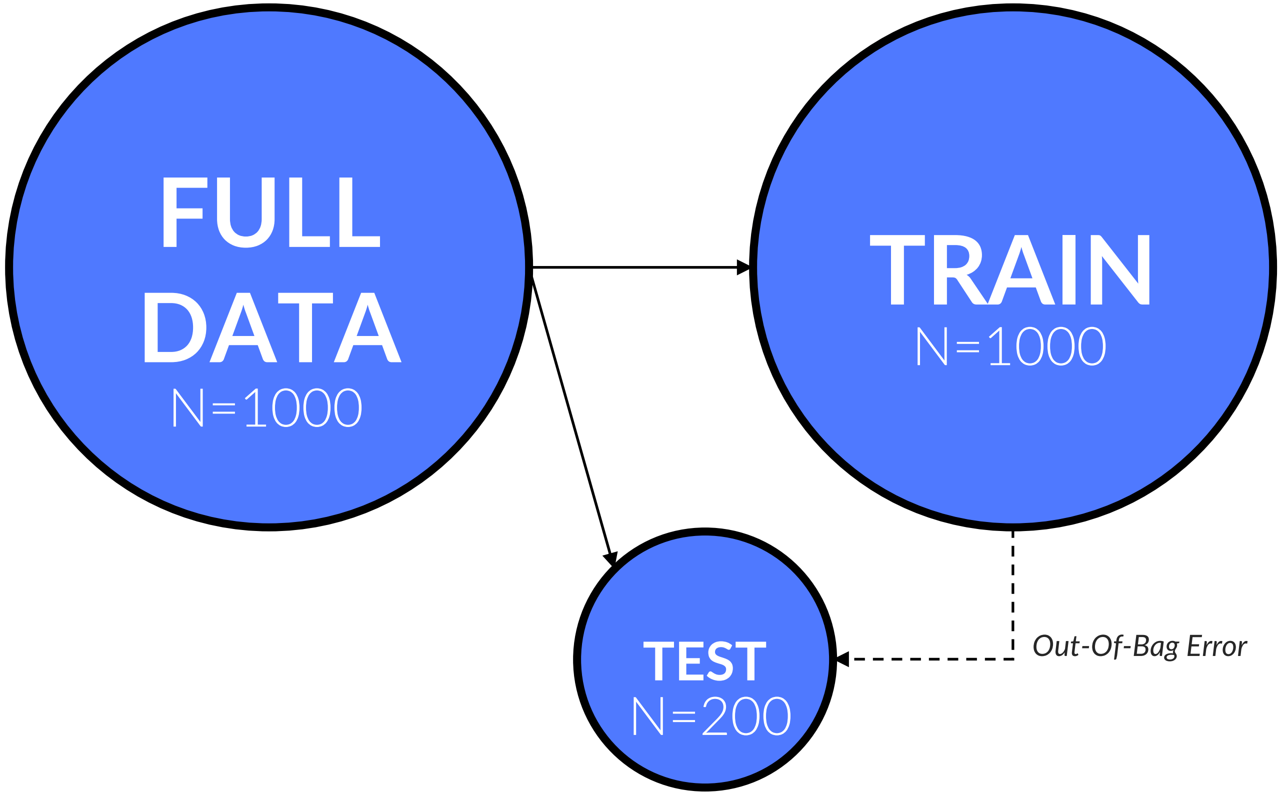
FULL DATASET

[1]	Train	Train	Train	Train	Test
[2]	Train	Train	Train	Test	Train
[3]	Train	Train	Test	Train	Train
[4]	Train	Test	Train	Train	Train
[5]	Test	Train	Train	Train	Train

People normally
do 5, 10 K-Fold

Bootstrap Sampling

(aka. Random sampling with replacement)

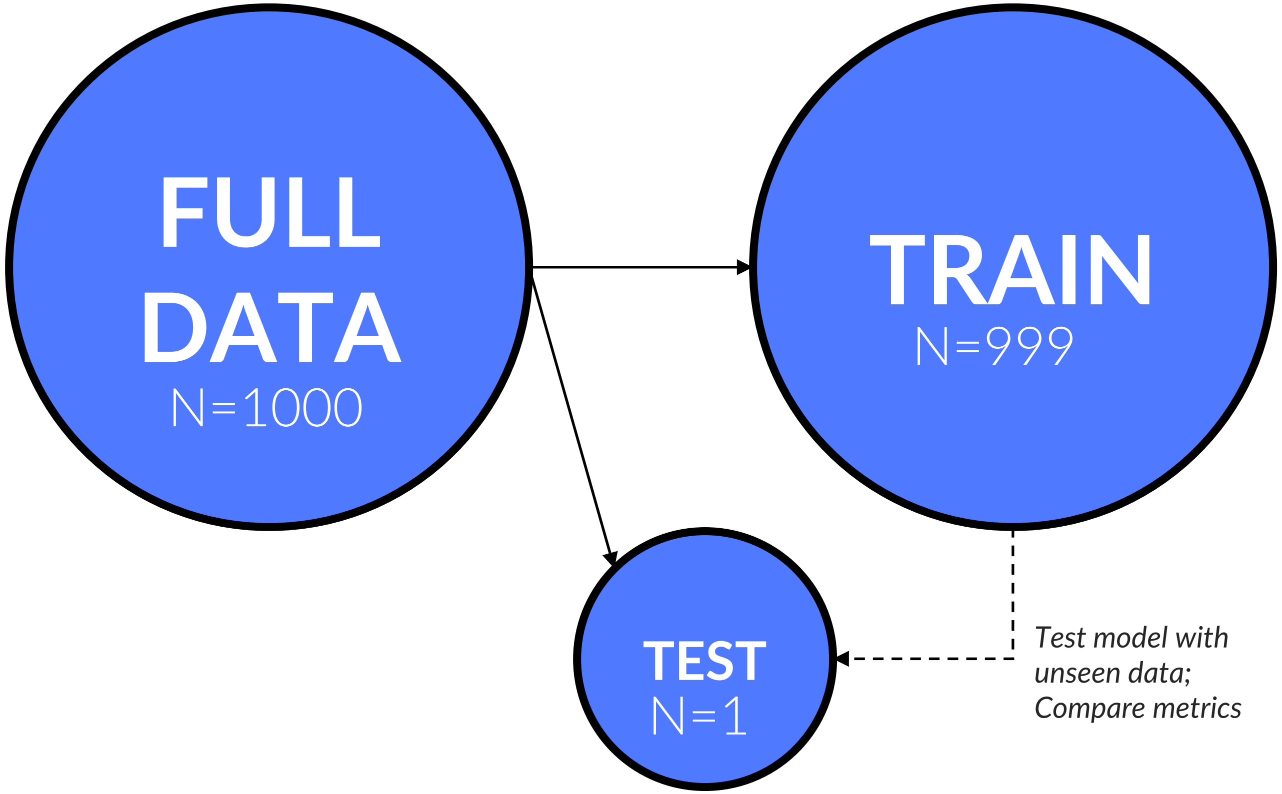


Explanation:

1000 may come from 800 unique data points + 200 repeated data points

Leave One Out CV

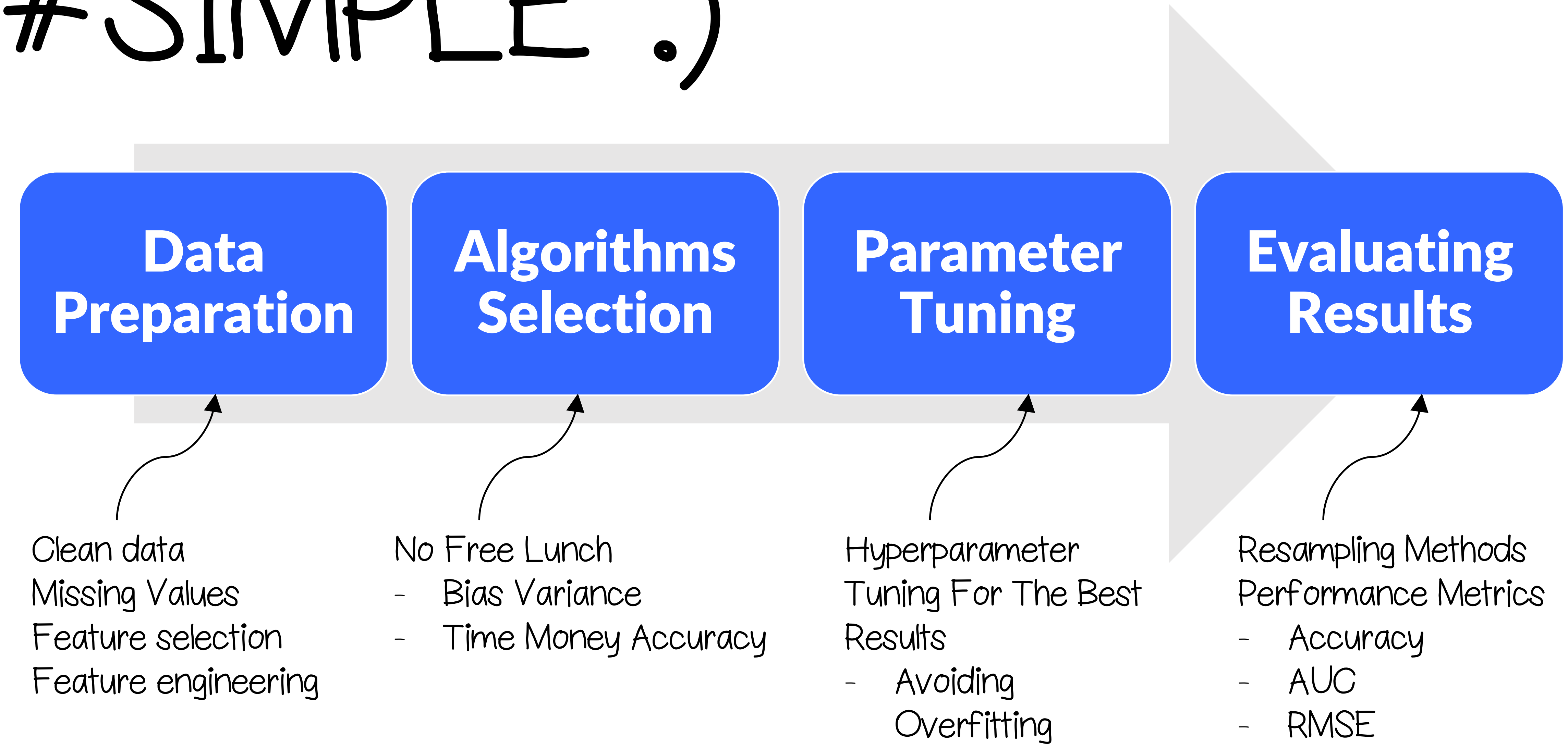
This is the most expensive method – time consuming



Explanation:

Leave 1 data point to test the model. For n=1000, this means we need to run model 1000 times

#SIMPLE :)



Data Preparation
Algorithms Selection
Parameter Tuning
Evaluating Results
Clean data
Missing Values
Feature selection
Feature engineering
No Free Lunch
- Bias Variance
- Time Money Accuracy
Hyperparameter Tuning For The Best Results
- Avoiding Overfitting
Resampling Methods
Performance Metrics
- Accuracy
- AUC
- RMSE

SUMMARY

- Data preparation takes 70-80% of our time.
    - “Garbage in, Garbage out”
    - Feature selection | Feature engineering | Missing Value
- No free lunch theorem
    - **Bias vs. Variance Tradeoff**
    - **Speed vs. Accuracy Tradeoff**
- [Hyper]Parameter tuning to get the best out of our learning algorithms
- Resampling techniques used in ML research incl. **train-test**, **k-fold CV**, **repeated k-fold CV**, bootstrap, and LOOCV

OKAY !! Let's recap a bit

- ML is function approximation
- Learn from past data to predict future (unseen) data point
- The goal is representation & generalization (avoid overfitting)
- Regression or Classification problems depend target variable (y)
- No Free Lunch: bias vs. variance tradeoff
- Algorithms can be tuned via hyperparameter
- K-Fold cross validation is popular resampling technique
- Our model passes the test if training vs. testing performances are similar (not overfit)

ALGORITHMS

YOU SHOULD KNOW

We cover 8 algorithms

Linear Regression	Regression
Logistic Regression	Classification
Decision Tree	Regression & Classification
Random Forest	*(mainly)* Classification
KNN	*(mainly)* Classification
Support Vector Machine	*(mainly)* Classification
Neural Networks	Regression & Classification
K-Means	Unsupervised Learning

Supervised Learning

NN is special algorithm, it works like reinforcement learning

LINEAR REGRESSION

Hello World in Machine Learning

Recall these steps

Data Preparation

Clean data
Missing Values
Feature selection
Feature engineering

Algorithms Selection

No Free Lunch
- Bias Variance
- Time Money Accuracy

Parameter Tuning

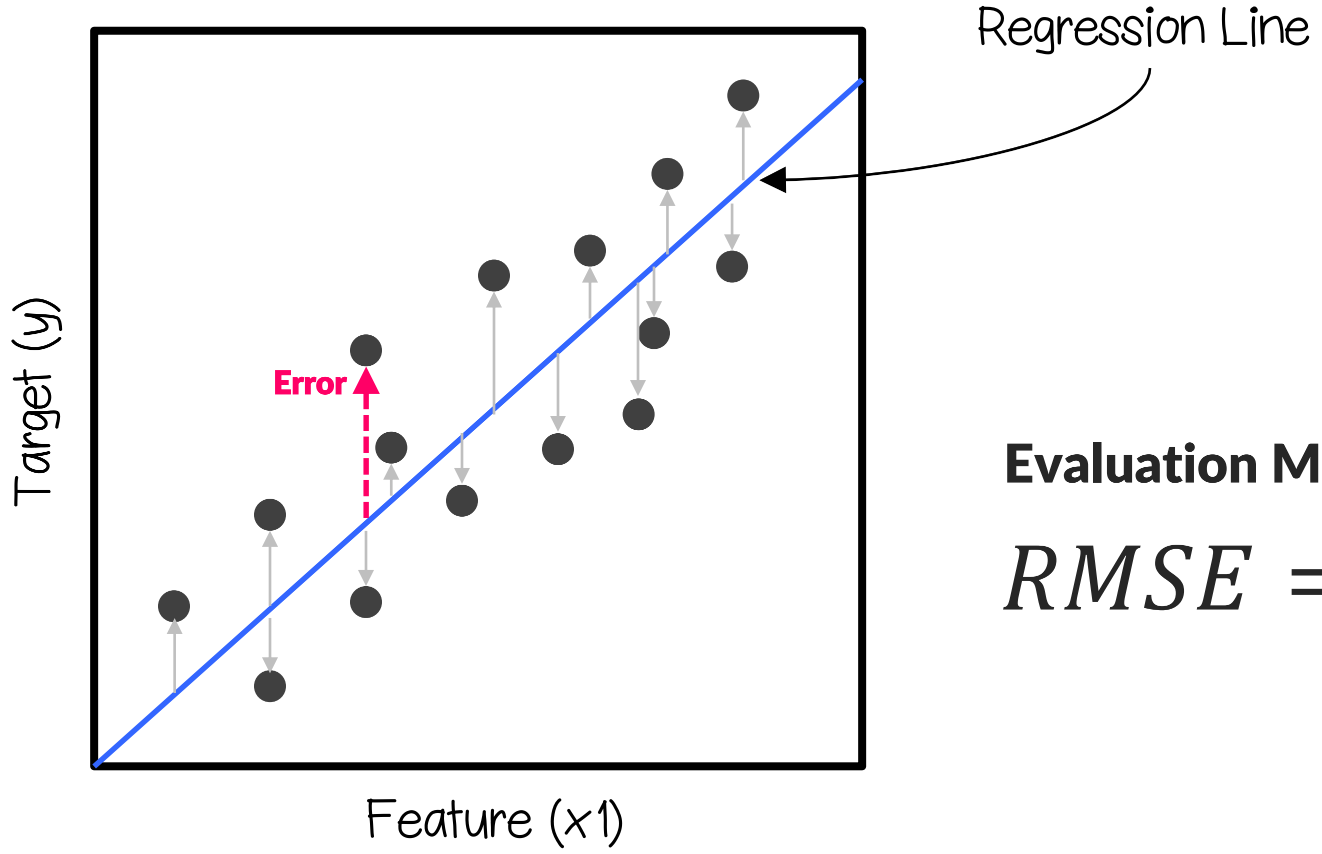
Hyperparameter Tuning For The Best Results
- Avoiding Overfitting

Evaluating Results

Resampling Methods
Performance Metrics
- Accuracy
- AUC
- RMSE

Algorithm explained:

How to draw a single straight line that fit best with given data points

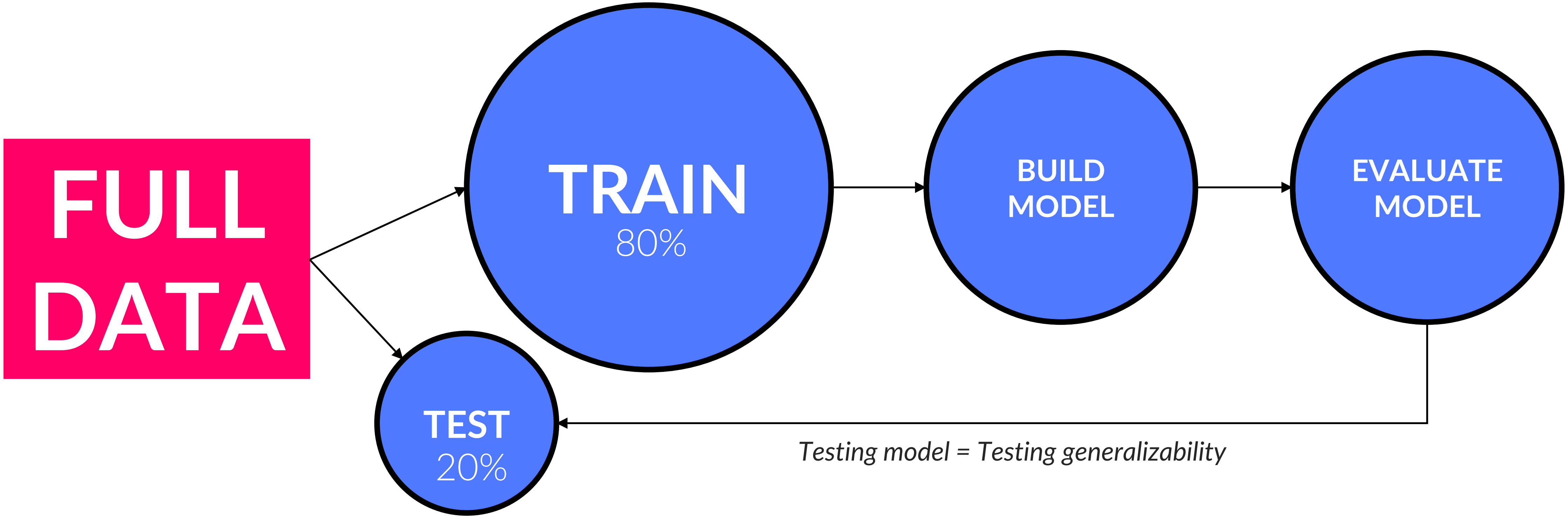


We want to
minimize RMSE
(cost function)

Evaluation Metric

$$RMSE = \sqrt{mean\ (error^2)}$$

Let's do a **train-test split** for linear regression problem



Advantages:

- *Easy to train*
- *Always find the optimum solutions (least squared)*
- *Foundation of advanced algorithms like neural nets*

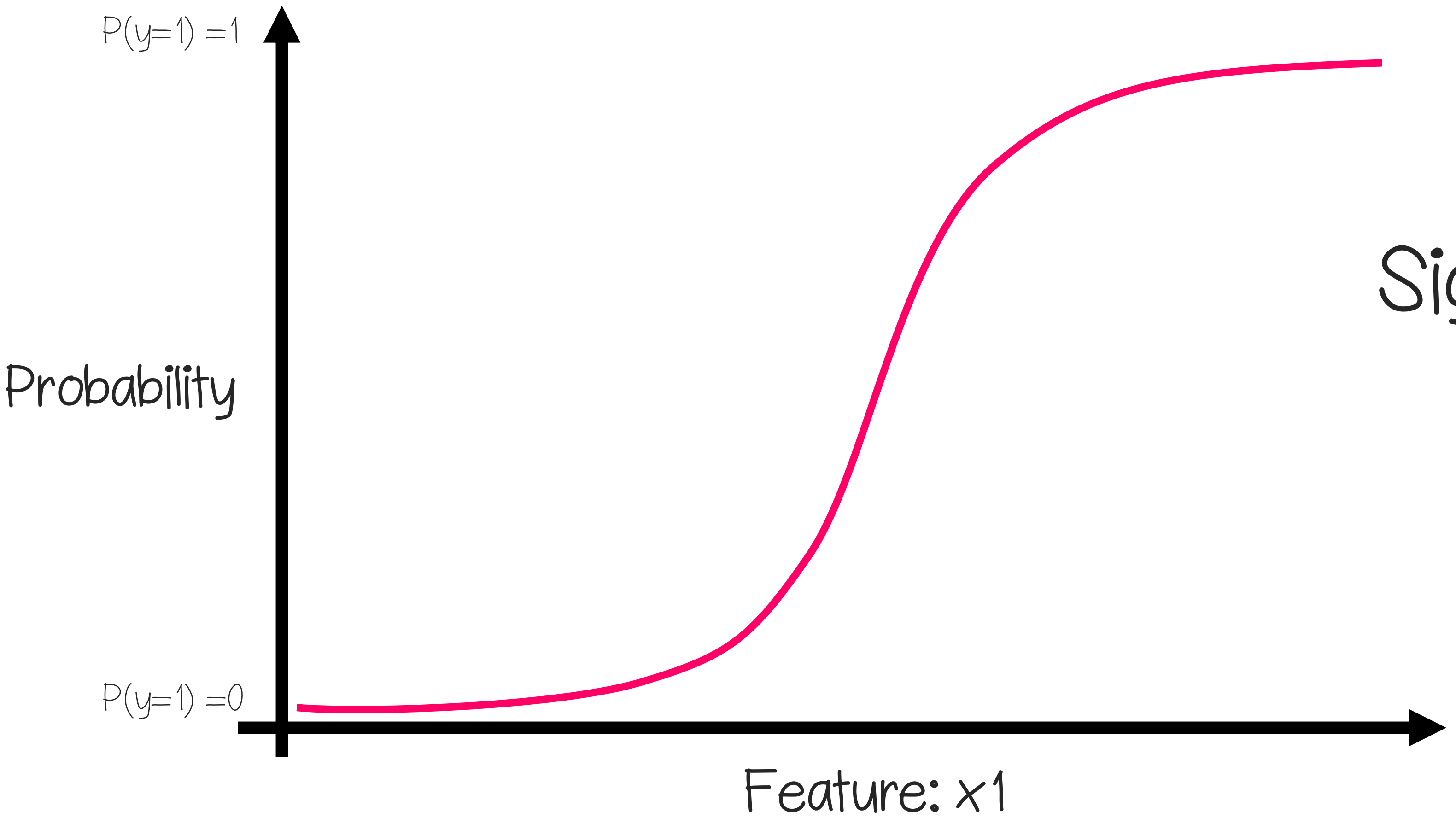
Disadvantages:

- *Only work for regression problems (continuous y)*
- *Full of assumptions (high bias algorithms)*

LOGISTIC REGRESSION

Algorithm explained:

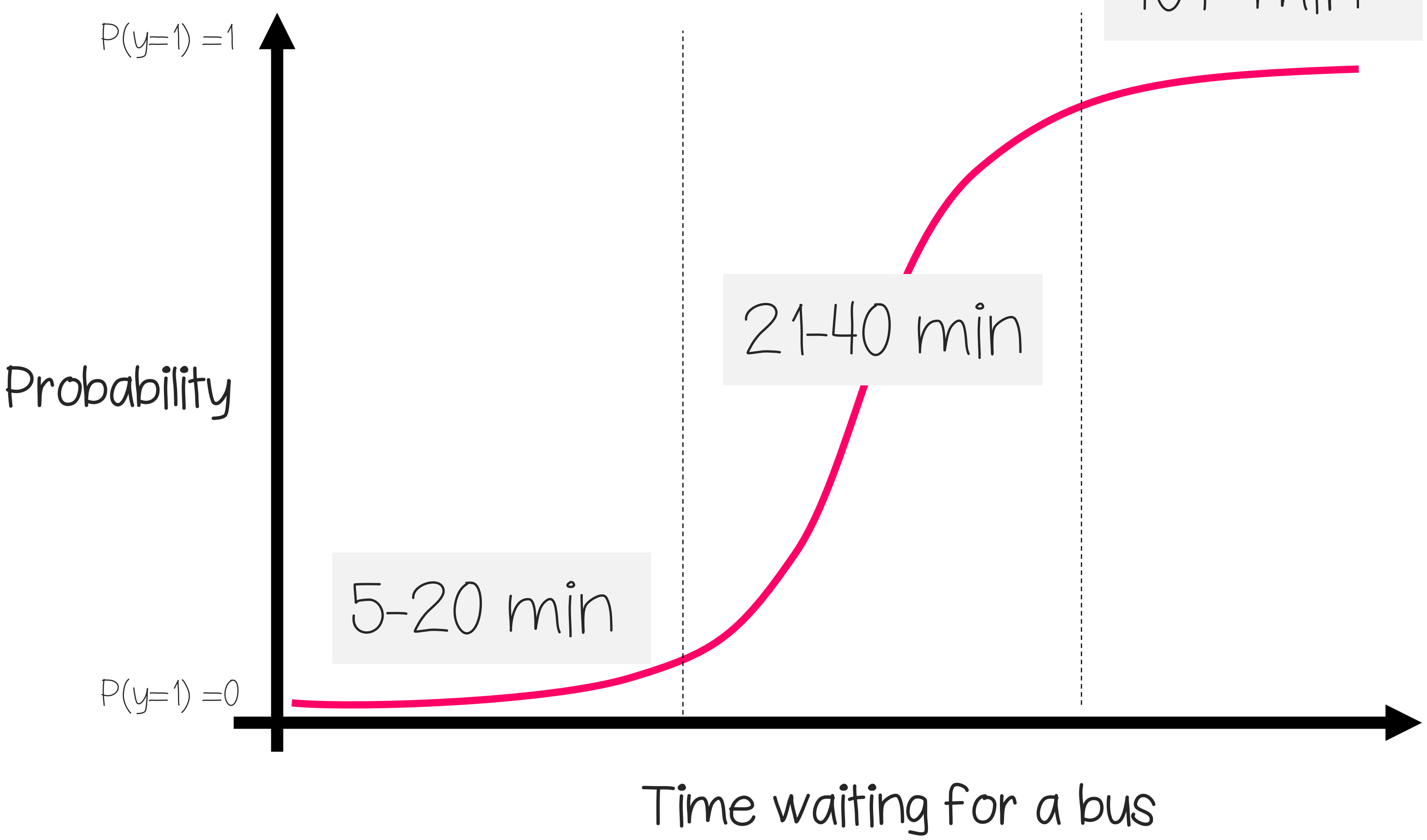
Classify data points by calculating the ***probability*** that Y=1



$$\frac{1}{1+e^{-z}}$$

Sigmoid Function

Should I get taxi or a bus?



The model is very similar to linear regression

(assuming linear relationship between x and y)

z = bo + b1x1 + b2x2 + b3x3 + b4x4 → then $1 / 1 + e^{-z}$ to get probability scores

ID	X1 (minute)	X2 (crowd)	X3 (time)	X4 (weather)	P(Y=1) (get Taxi)	Prediction
1	50	Many	Friday	Good	.95	Yes
2	20	Many	Saturday	OK	.32	No
3	20	Few	Monday	OK	.31	No
4	25	Few	Friday	Bad	.85	Yes
5	30	Many	Friday	Good	.68	Yes
6	35	Few	Saturday	OK	.25	No
7	12	Many	Monday	Good	.52	Yes
8	5	Few	Saturday	OK	.12	No
9	10	Many	Friday	Bad	.20	No
10	65	Few	Friday	Good	.98	Yes

We set threshold = 0.5
If p(y=1) >= 0.5, then we
predicted YES get taxi!

Advantages:

- *One of the most powerful algorithms to date*
- *Easy to train*
- *Can be developed to create complex decision boundaries (e.g. x^2 y^2)*

Disadvantages:

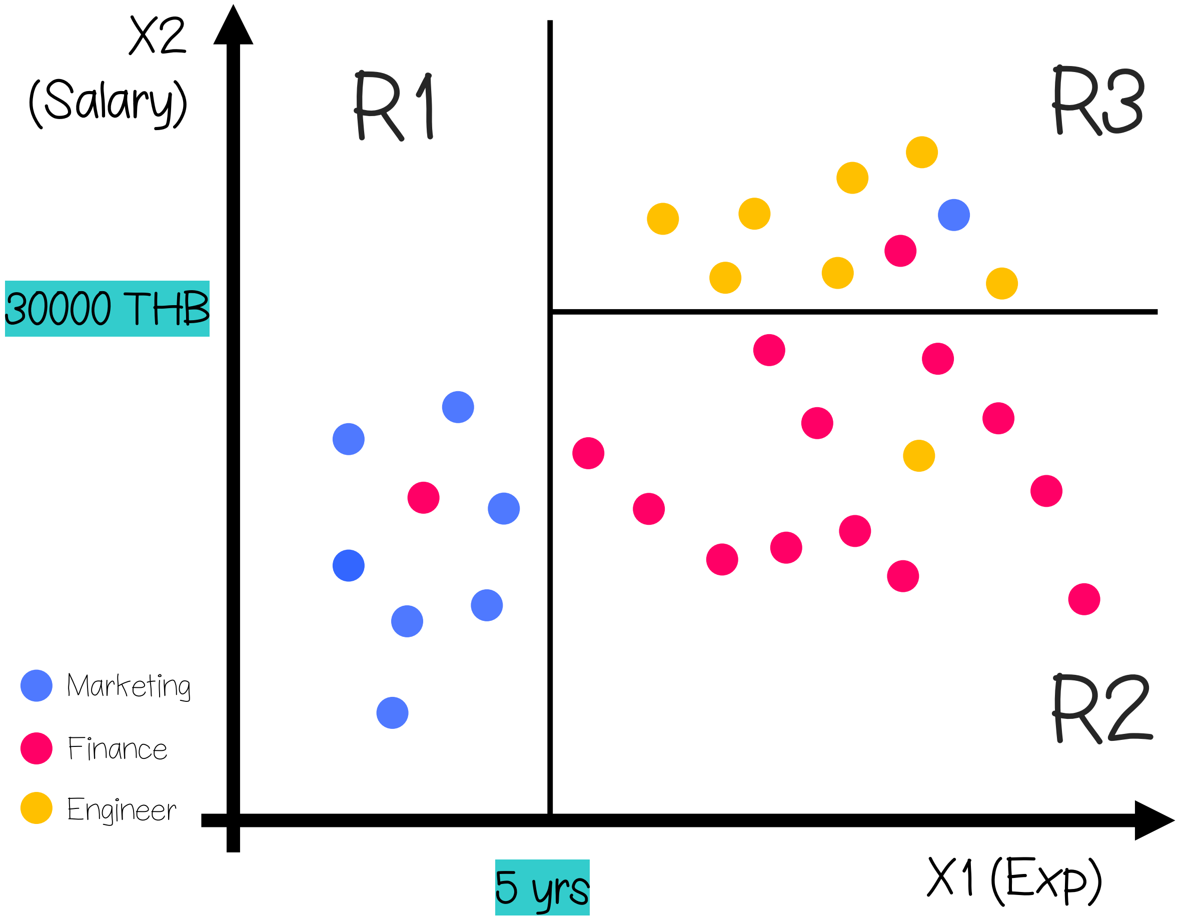
- *Also a lot of assumption e.g. no outlier, no noise*
- *Logistic regression is also a linear algorithm*
- *Prone to multicollinearity (check correlation first)*

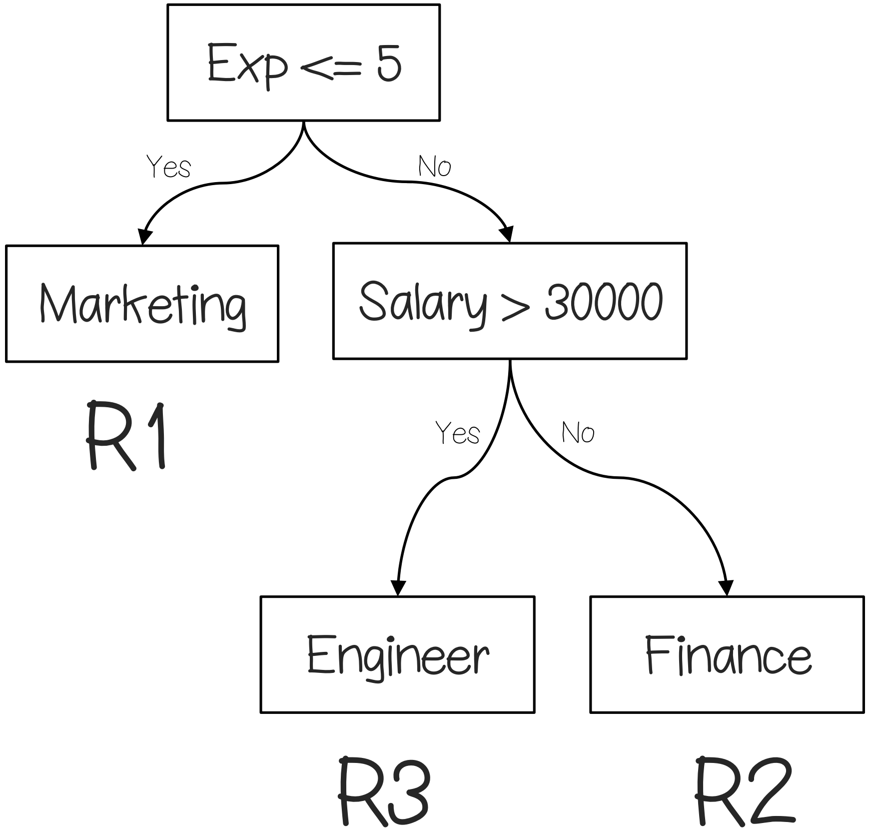
(sigmoid function transform an output into non-linear)

DECISION TREE

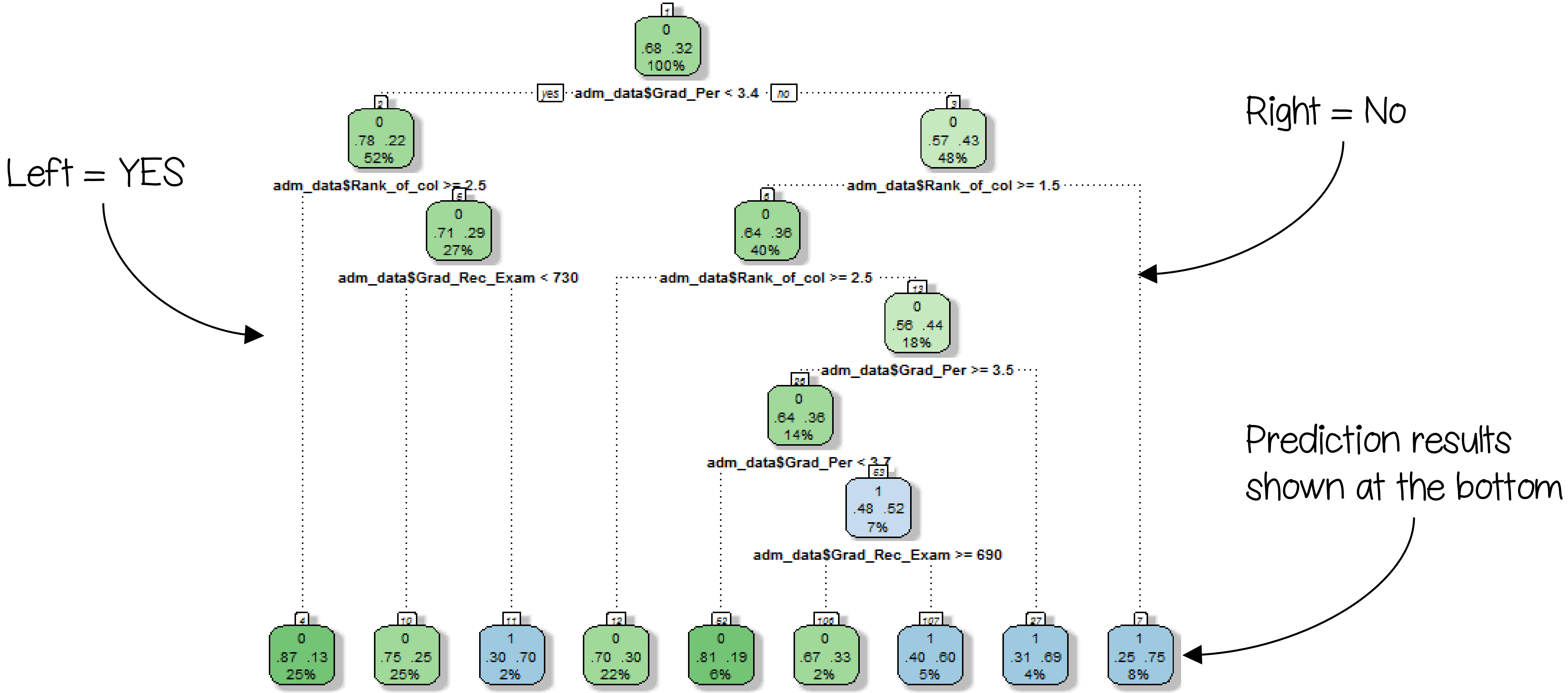
Algorithm explained:

Divide space into regions that best predict data points (classes)





Decision Tree is easily presented using diagram



Advantages:

- *Easy to train*
- *Easy to interpret results*
- *Widely used in many disciplines*

The main reason why people use decision tree

Disadvantages:

- *Prone to overfitting: Greedy algorithms*

Locally optimal decision are made at each node, but can't guarantee globally optimal decision trees

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

Required Packages:

library(**caret**) – train & test models

library(**caTools**) – split data into training & testing sets

library(**rpart**) – train decision tree (CART): recursive partitioning and regression tree

CART

```
method = 'rpart'
```

Type: Regression, Classification

Tuning parameters:

- `cp` (Complexity Parameter)

Required packages: `rpart`

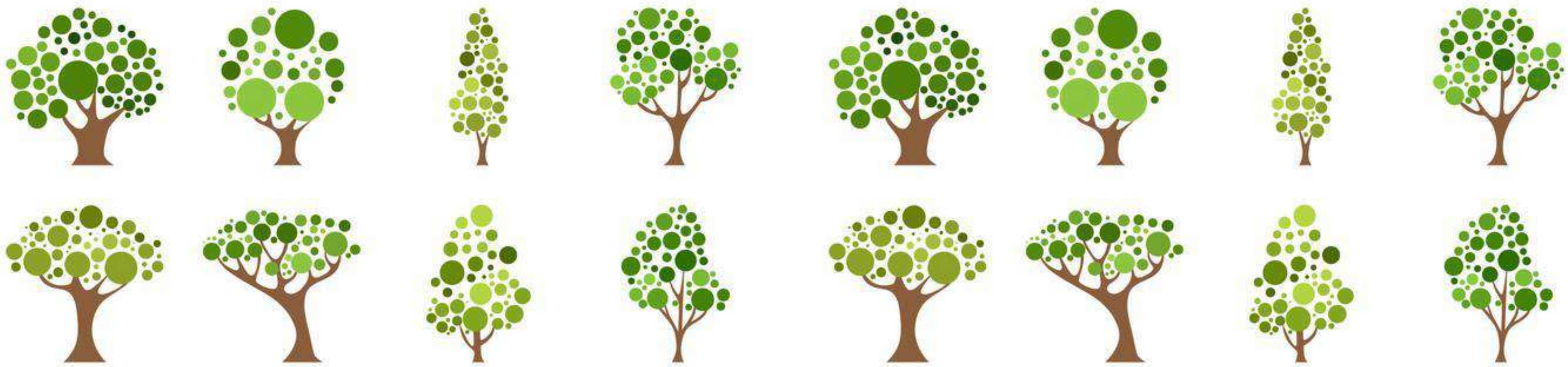
A model-specific variable importance metric is available.

Cp: complexity parameter to penalize larger trees

RANDOM FOREST

Algorithm explained:

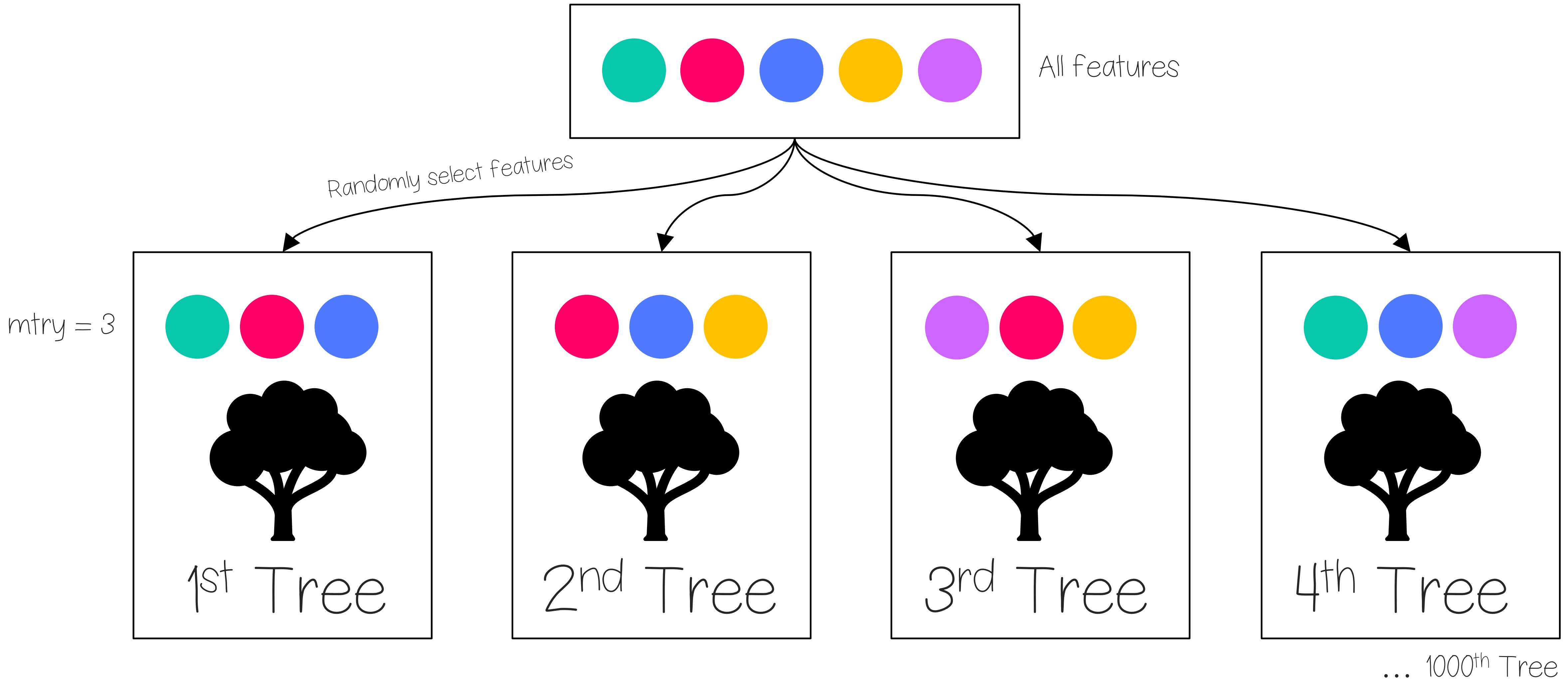
Grow thousand trees and average their prediction results



*Grow **uncorrelated** 1000 trees*

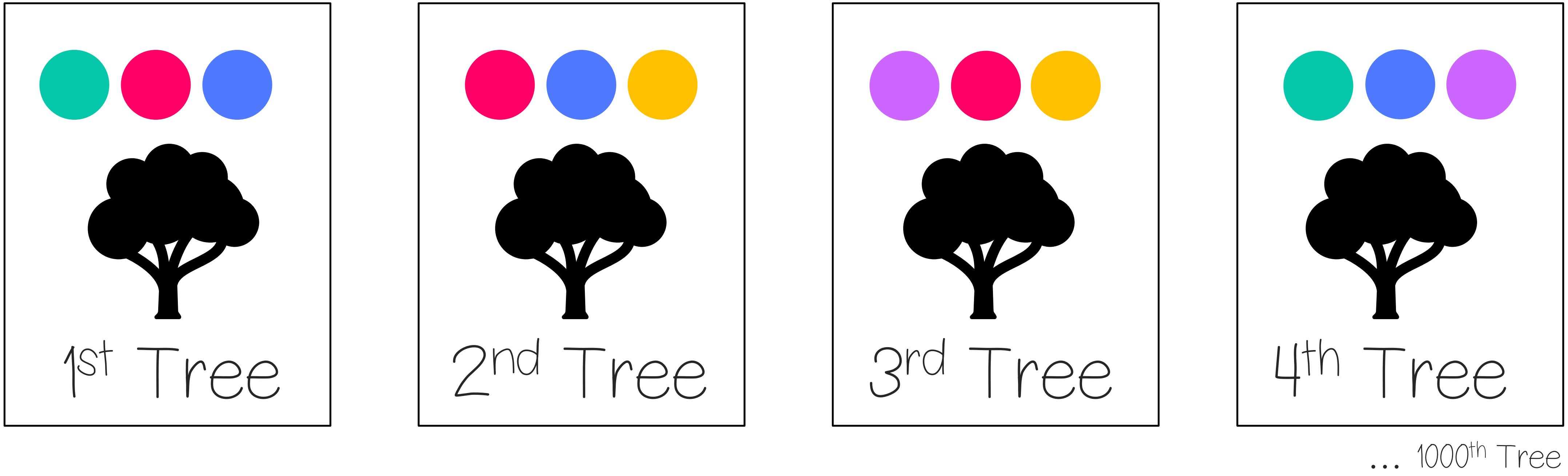
How do we grow 1000 trees?

Randomness is your friend, hyperparameter is mtry



Wisdom of the crowds

Find the average prediction results of all grown trees



Advantages:

- *Amongst the best learners in the world*
- *Easy to train (Improvement from Decision Tree)*

Disadvantages:

- *Can be slow to train for large dataset*
- *Black Box Algorithm (not easy to explain)*

Required Packages:

library(**caret**) – train & test models

library(**caTools**) – split data into training & testing sets

library(**randomForest**) – train random forest in caret

Random Forest

```
method = 'rf'
```

Type: Classification, Regression

Tuning parameters:

- `mtry` (#Randomly Selected Predictors)

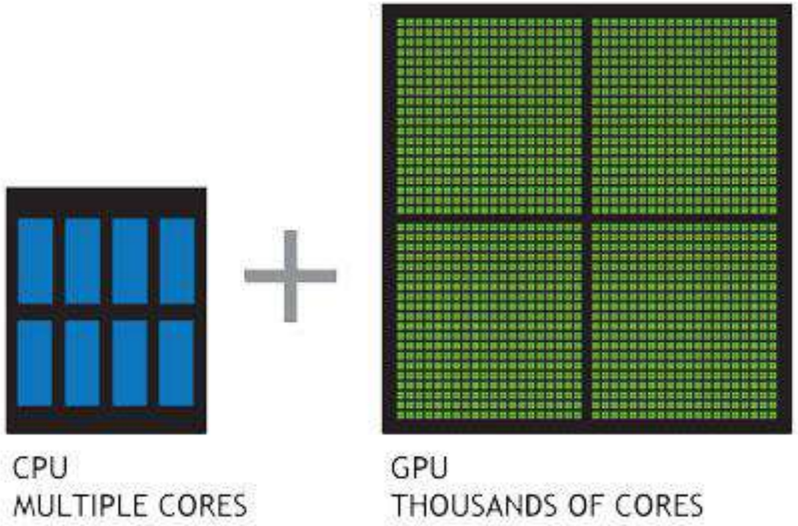
Tune mtry parameter to find the best prediction from all trees

Required packages: `randomForest`

A model-specific variable importance metric is available.

Let’s talk about the hardware side

Graphic Cards have GPU 1000 cores

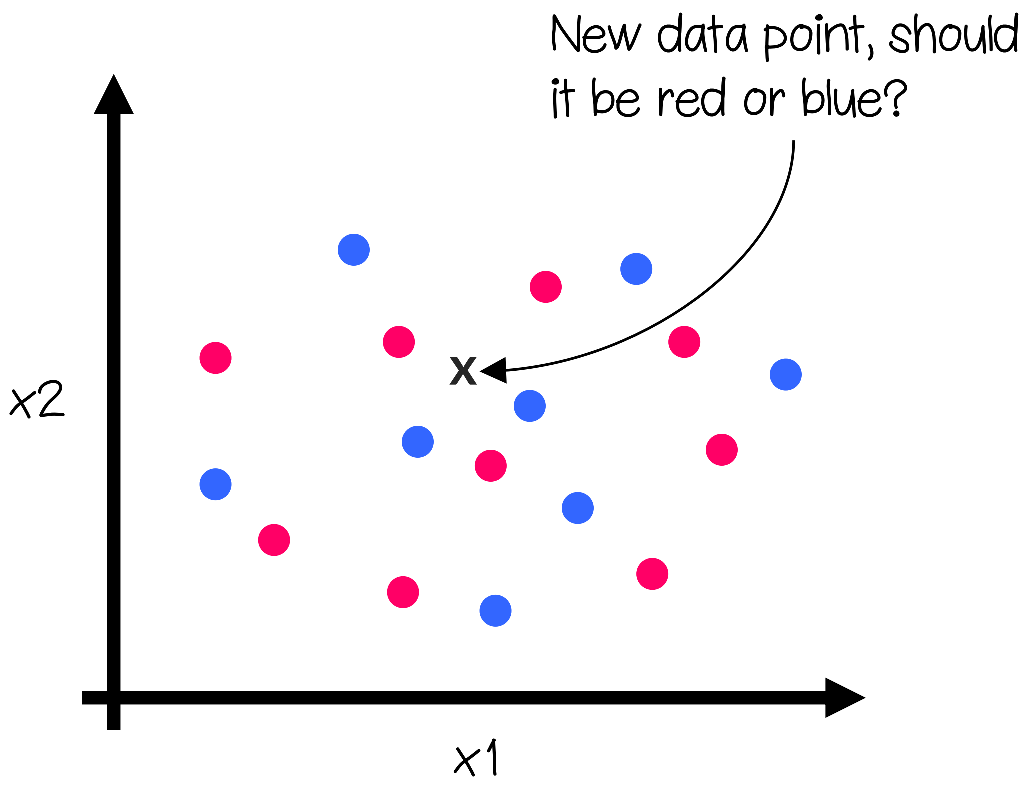


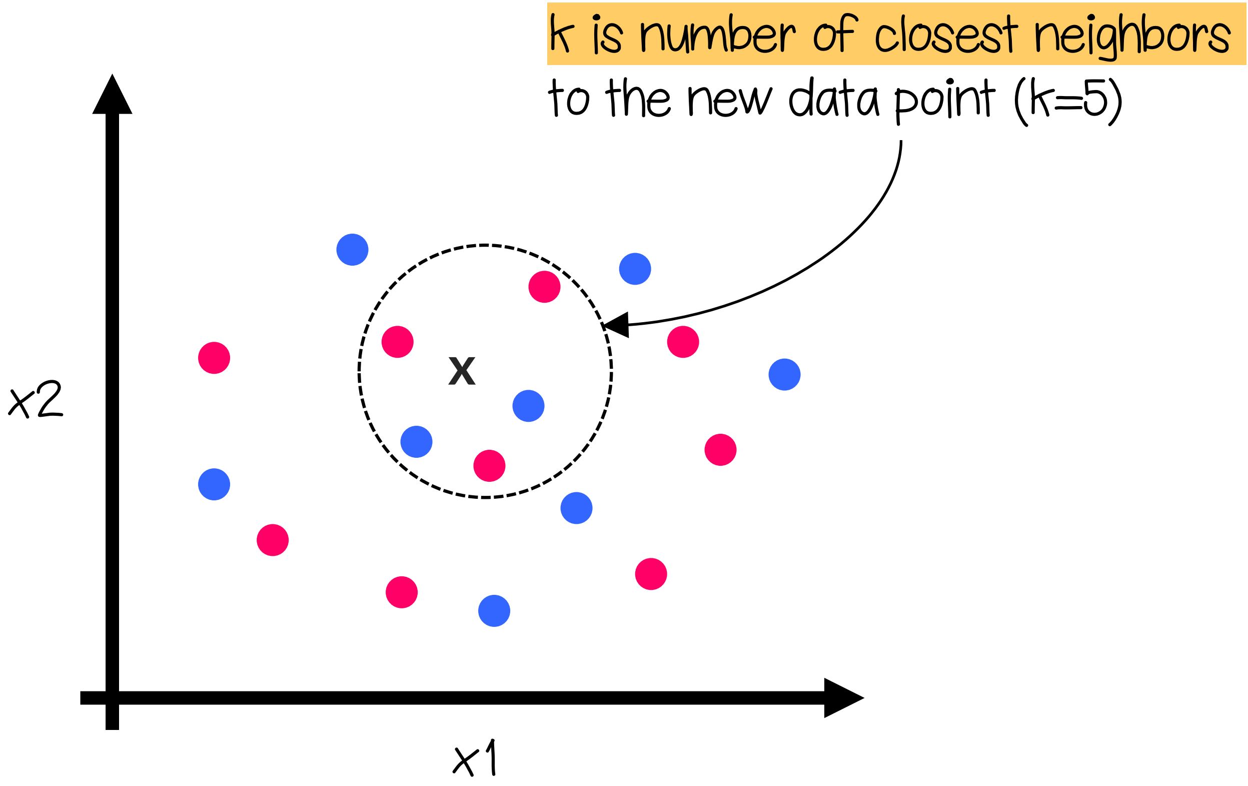
Nvidia has CUDA technology
Read more: https://developer.nvidia.com/about-cuda

KNN
(K-NEAREST NEIGHBOR)

Algorithm explained:

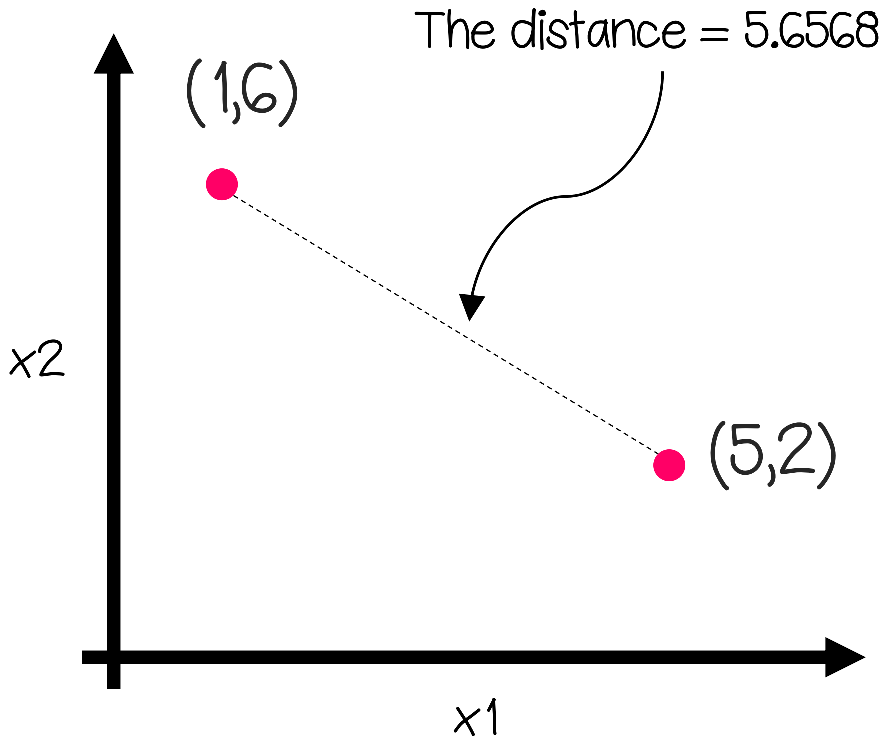
Predict new data according to the distance from its neighbors





Euclidean Distance:

The most popular distance calculation method



$$d = \sqrt{\sum (x_i - y_i)^2}$$

$$d = \sqrt{(1-5)^2 + (6-2)^2}$$

$$d = \sqrt{16 + 16}$$

$$\boldsymbol{d = 5.656854}$$

Very simple to compute

Another Example

Consider the `nutrient` dataset provided with the `flexclust` package. The dataset contains measurements on the nutrients of 27 types of meat, fish, and fowl. The first few observations are given by

```
> data(nutrient, package="flexclust")
> head(nutrient, 4)

                 energy protein fat calcium iron
BEEF BRAISED        340      20  28       9  2.6
HAMBURGER           245      21  17       9  2.7
BEEF ROAST          420      15  39       7  2.0
BEEF STEAK          375      19  32       9  2.6
```

We can calculated many dimensions (more than two)

and the Euclidean distance between the first two (beef braised and hamburger) is

$$d = \sqrt{(340-245)^2 + (20-21)^2 + (28-17)^2 + (9-9)^2 + (26-27)^2} = 95.64$$

Advantages:

- *Very simple*
- *Easy to train (remember all data)*

Disadvantages:

- *Worst for large dataset*
- *Not good with high dimensional data (many features)*
- *Don’t work well with categorical data*

Example Code:

```r
## load library
library(caret)

## train model
## define trControl
my_trcontrol = trainControl(method = "cv",
                            number = 3,
                            verboseIter = TRUE,
                            search = "random")

knn_model <- train(Species ~ . , data = m,
                   method = "knn",
                   metric = "Accuracy",
                   tuneLength = 10,
                   trControl = my_trcontrol)
```

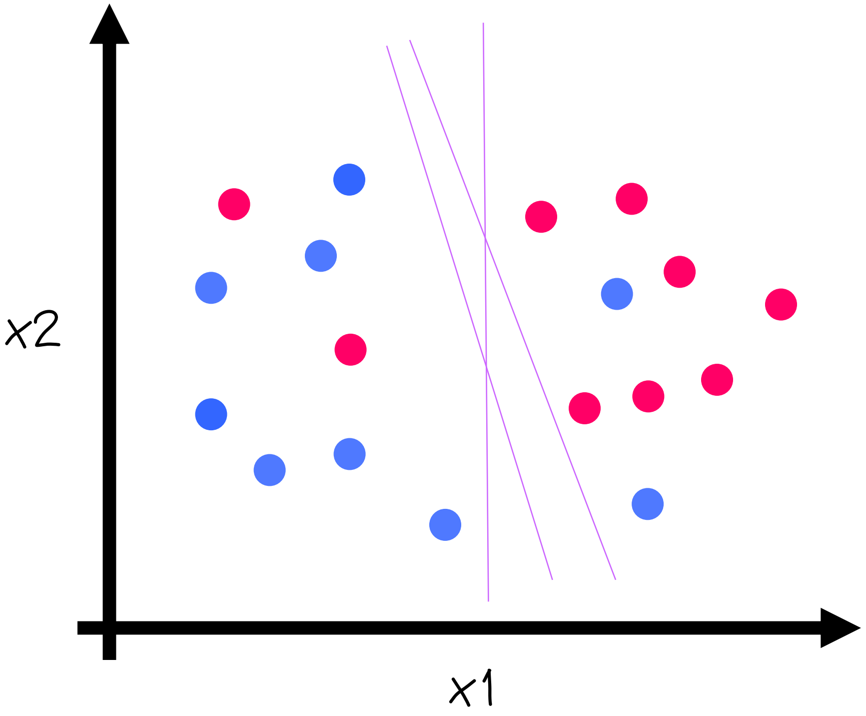
Define our technique
(to train the model)

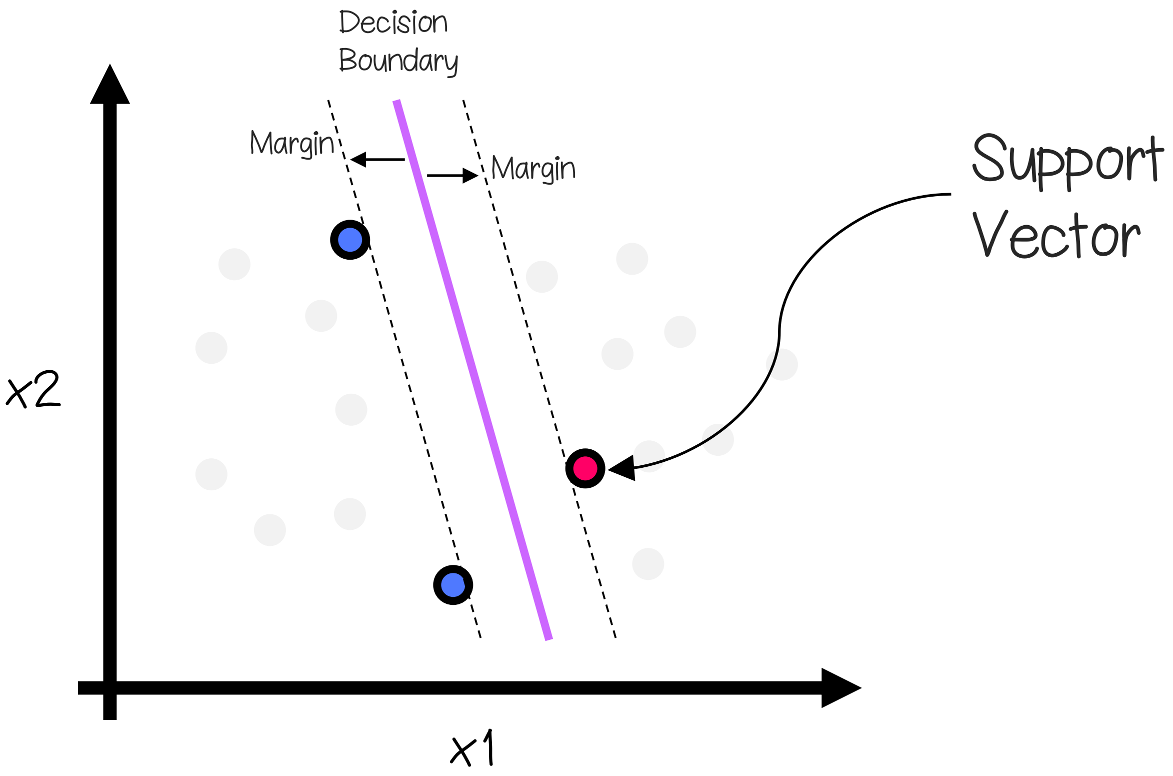
Train the model using
random search +
tuneLength = 10

SUPPORT VECTOR MACHINE

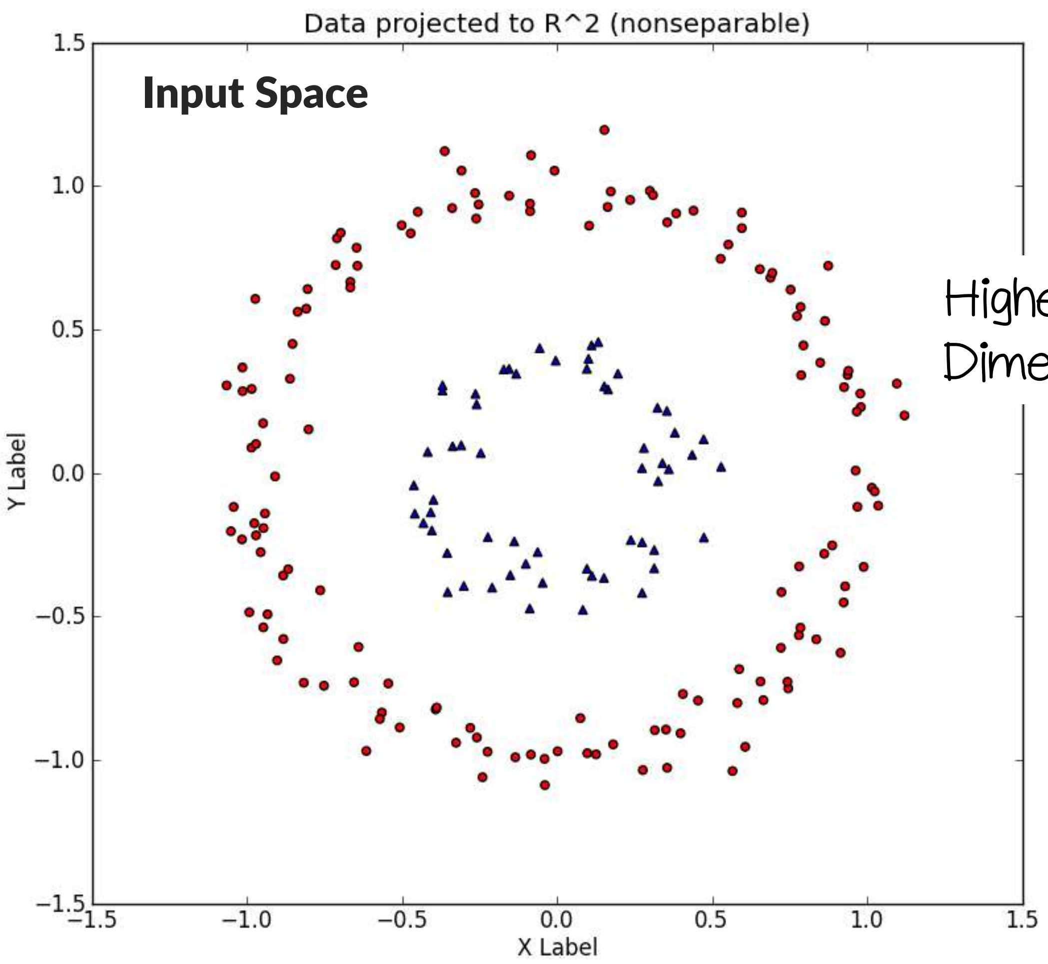
Algorithm explained:

We want to maximize the ***margin*** from decision boundary

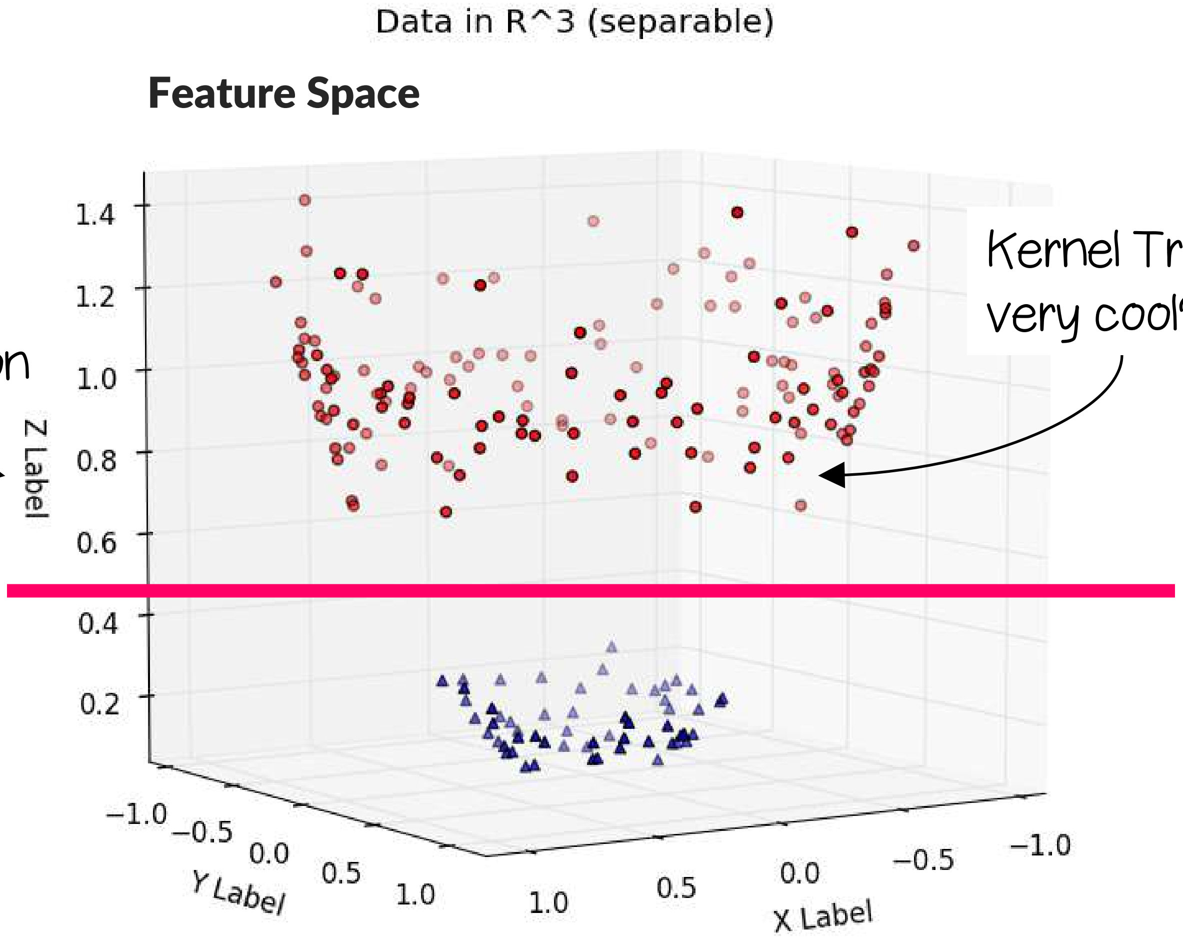




But can you divide this with a single straight line?

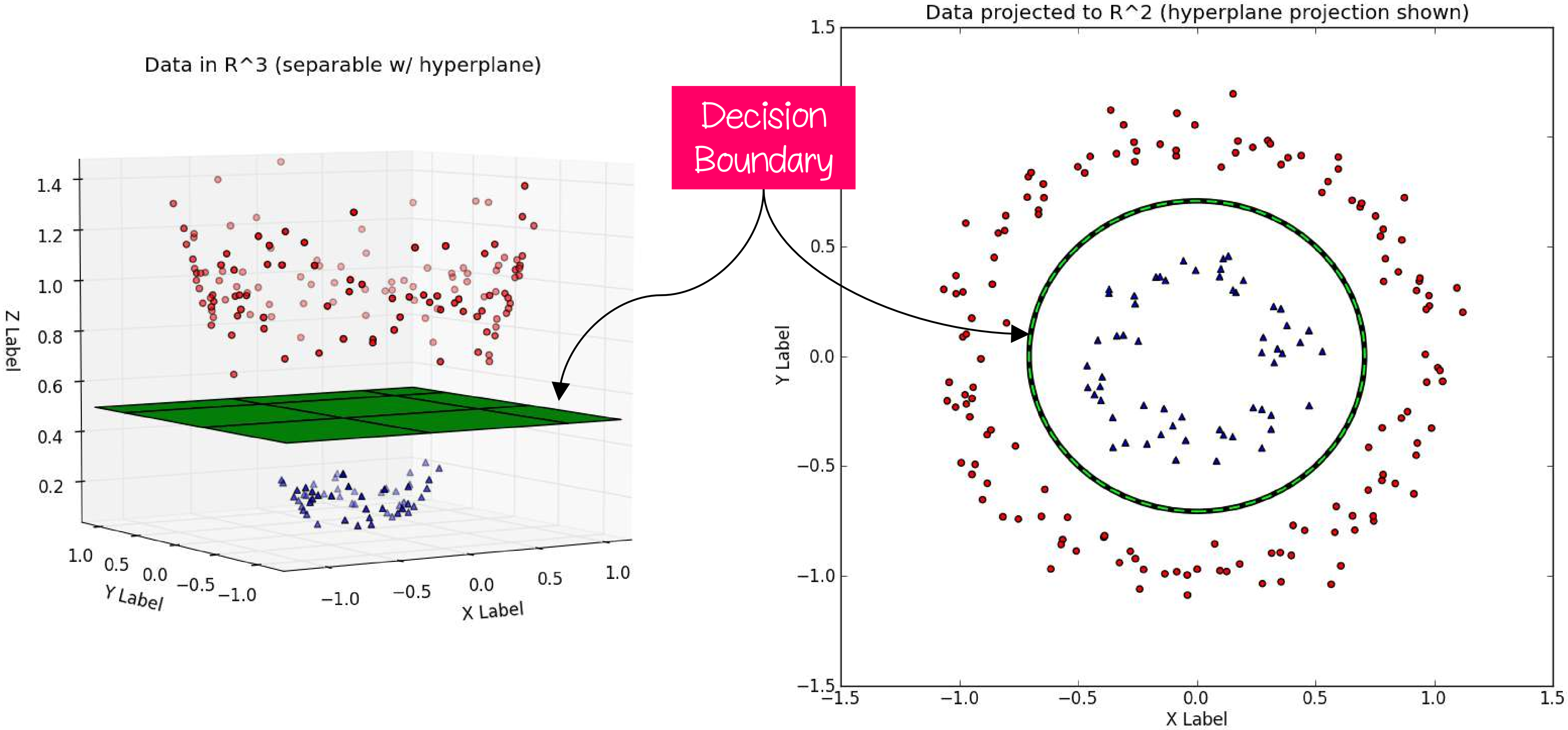


Higher Dimension



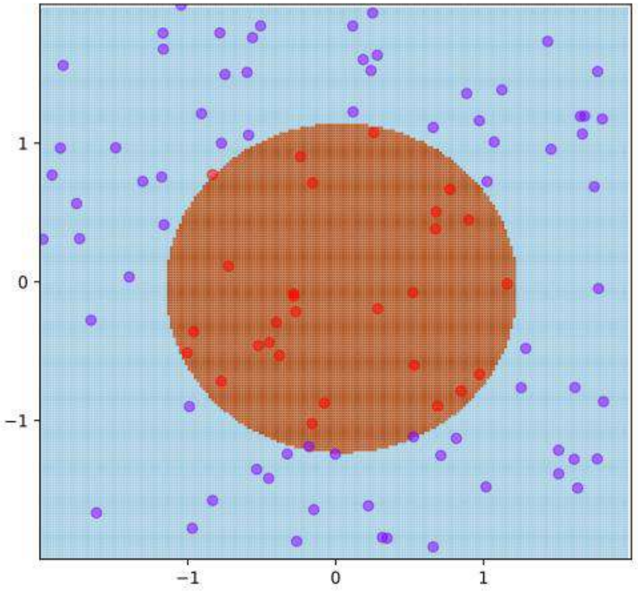
Kernel Trick, very cool?

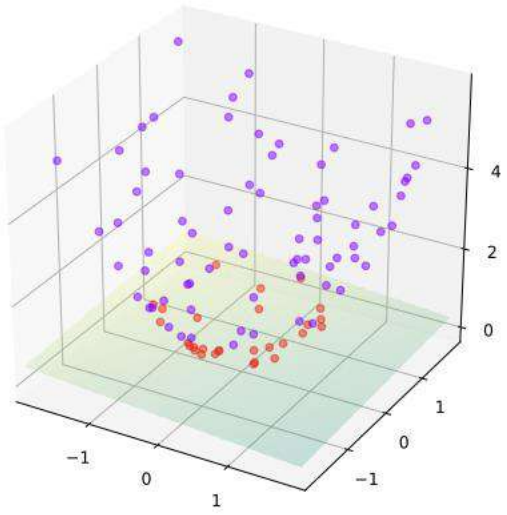
Impressive Result!!



Let's look closely at kernel trick

- *we have (x,y) coordinates*
- *we project a new higher dimension:* $$(x, y, \underbrace{x^2 + y^2}_{\text{3rd dimension}})$$





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

Advantages:

- *Work with Regression | Classification problems*
- *Superior Performance*
- *Having lots of available kernel tricks*

  https://topepo.github.io/caret/train-models-by-tag.html#support-vector-machines

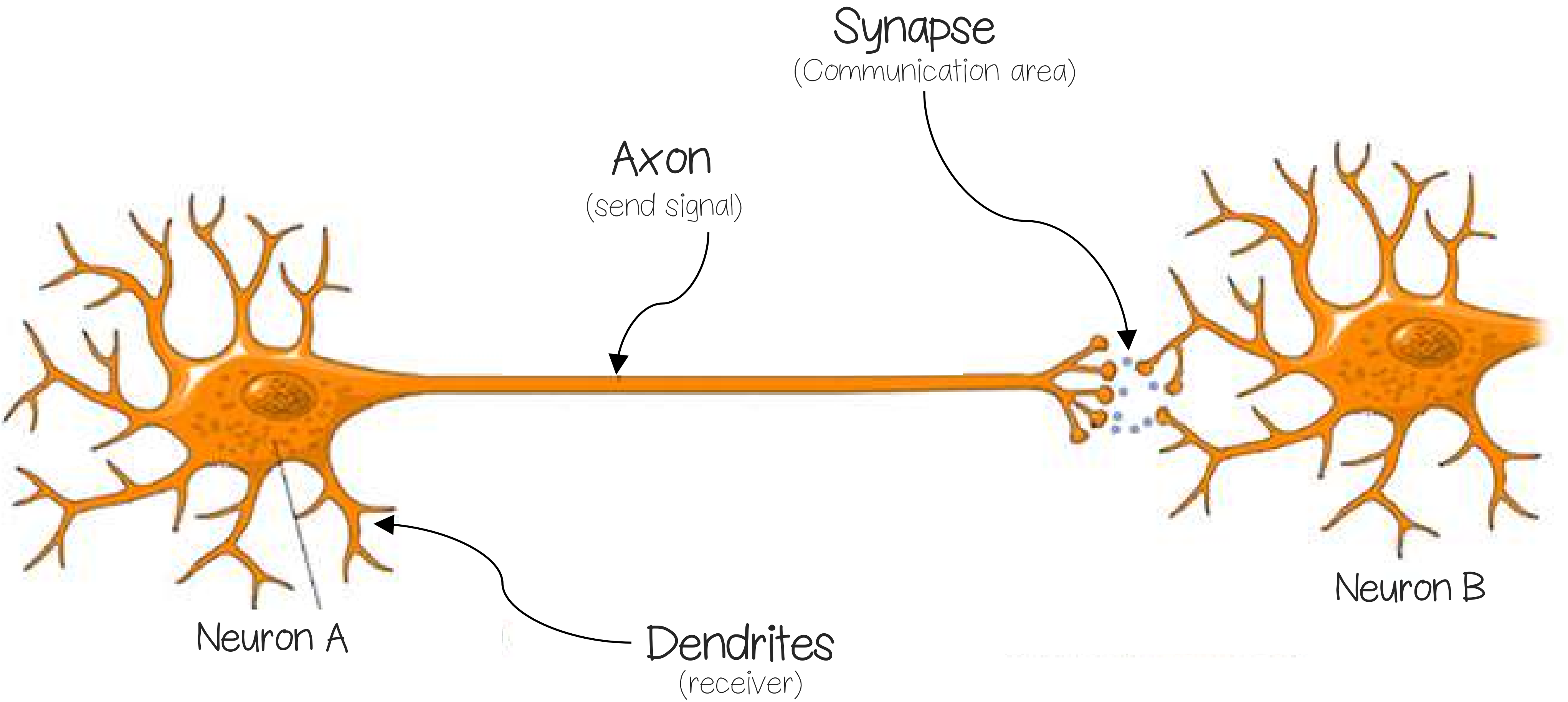
Disadvantages:

- *Expensive*
- *It’s lazy algorithm (remember data like KNN)*

NEURAL NETWORK


Neuron

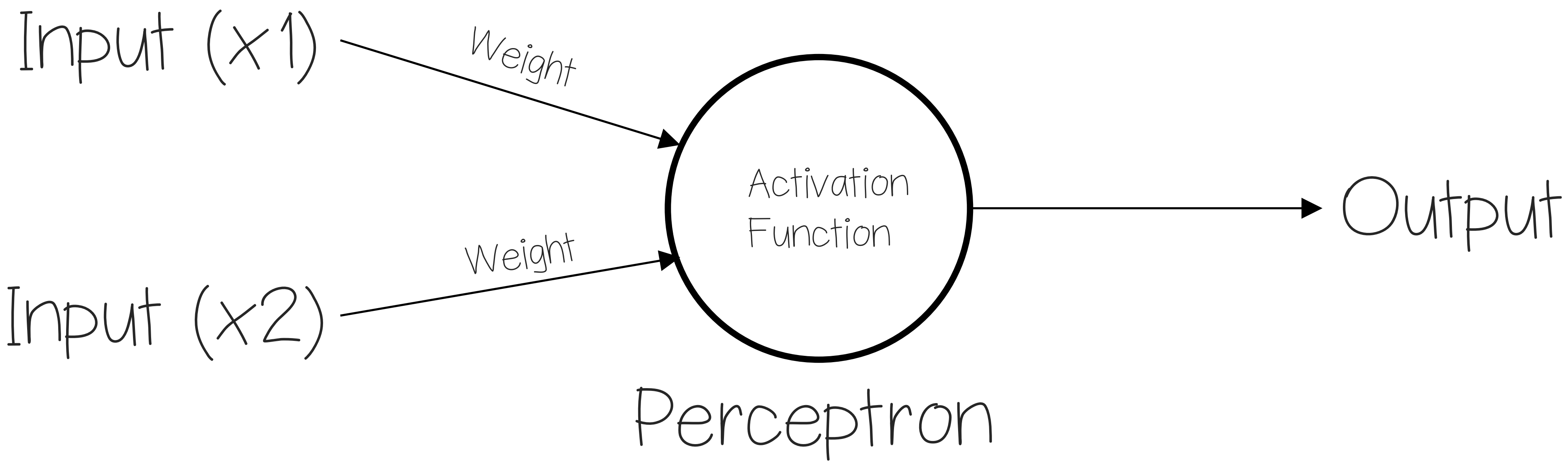
How your brain cells communicate



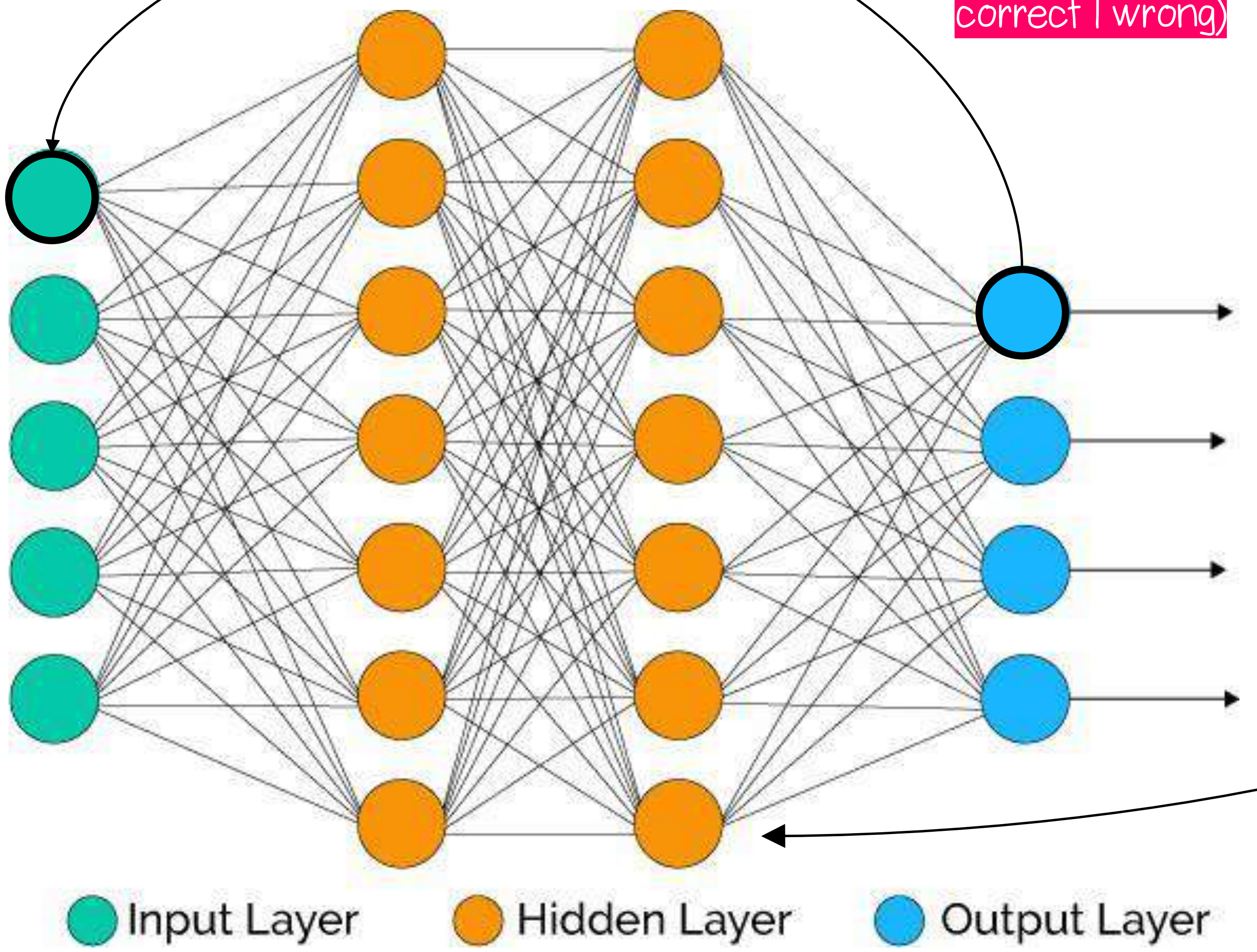

If communicated successfully,
the neuron fire!

Algorithm Explained:

Neurons fire only when it's the signal strong enough to activate them



Loss layer: send feedback to input layer (whether prediction is correct / wrong)



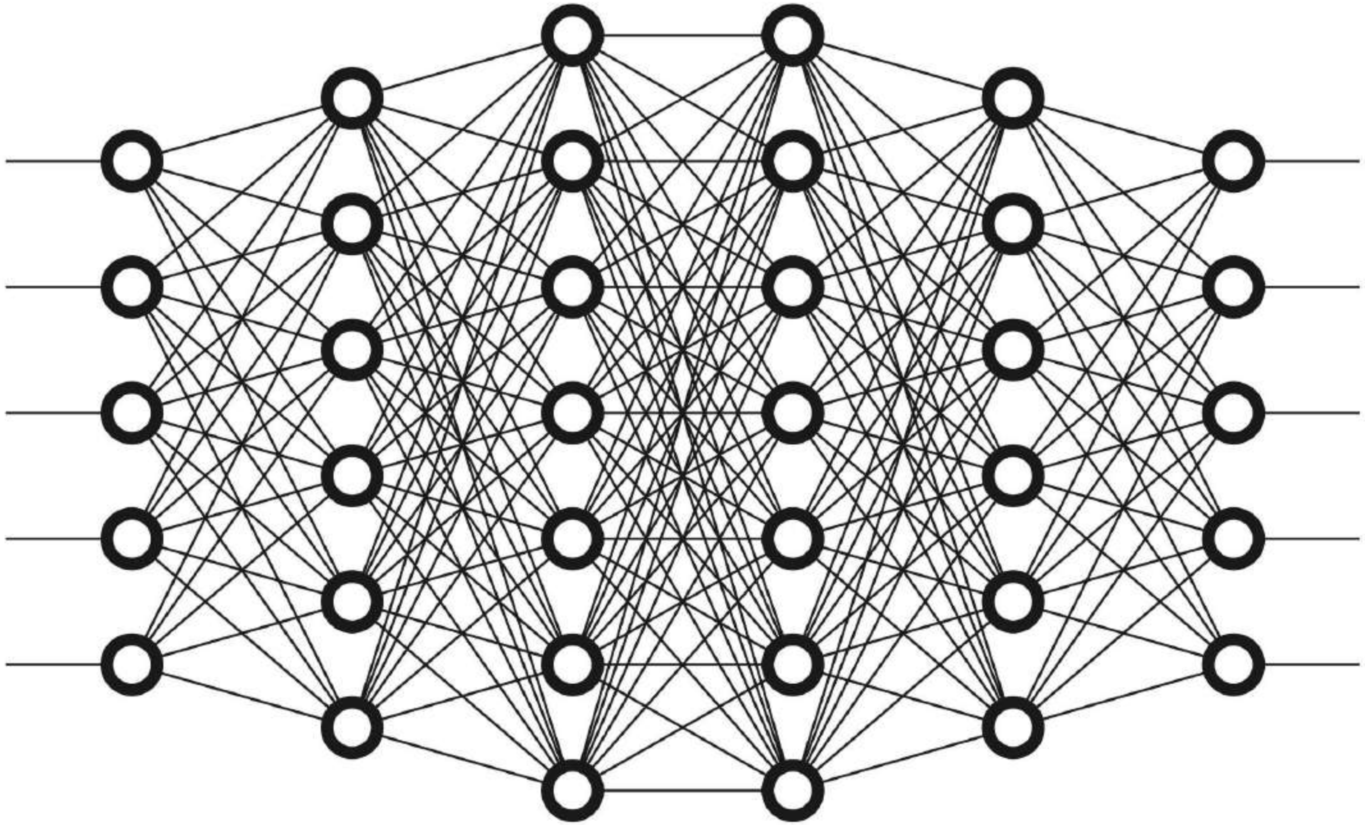
For basic neural networks, hyperparameter is number of hidden layer and learning rate (alpha)

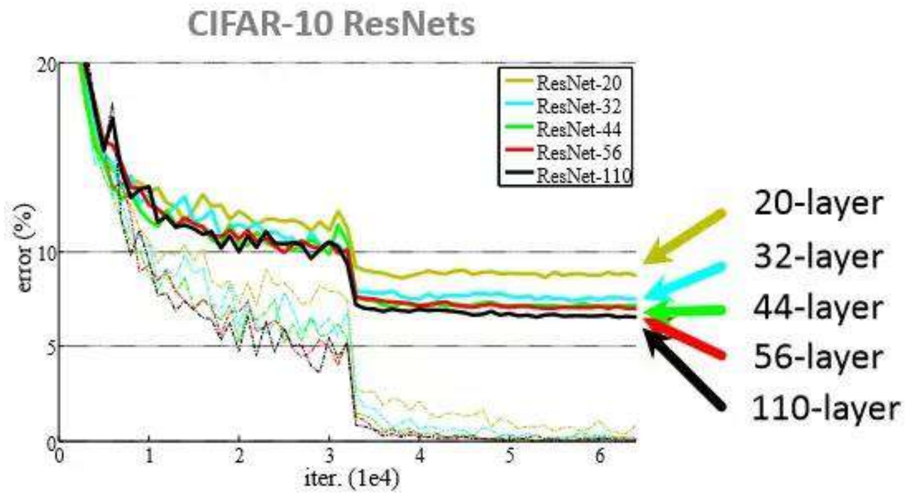
https://medium.com/@xenonstack/overview-of-artificial-neural-networks-and-its-applications-2525c1addff7

Remarks:
NN is a good example of supervised learning that also a reinforcement learning (learn from experience)

Deep Learning

Expands the number of hidden layers





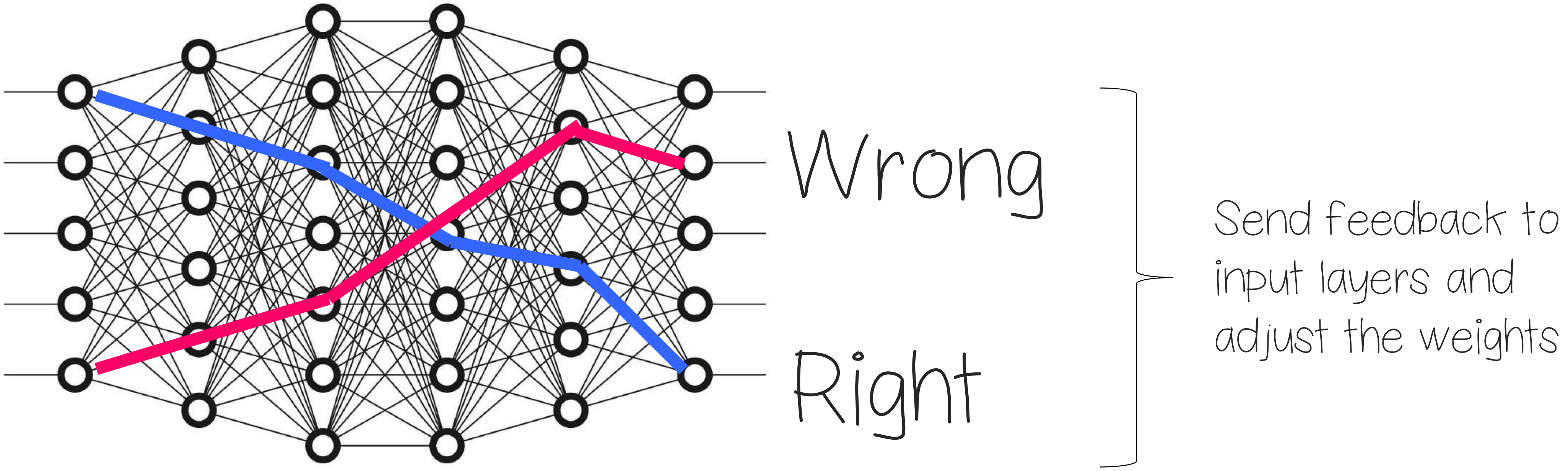
Hieu Pham, Has done some machine learning
Updated Apr 29, 2017

In this paper (Huang et al, 2016), the authors report that they "can increase the depth of residual networks ***even beyond 1200 layers*** and still yield meaningful improvements in test error (4.91% on CIFAR-10)".

I have never heard of anything deeper than that.

Backpropagation

The backbone of artificial neural networks



NO FREE LUNCH

- Speed
- Accuracy
- Interpretability
- Bias & Variance Tradeoff

ความสามารถในการ
อธิบายผลโมเดล

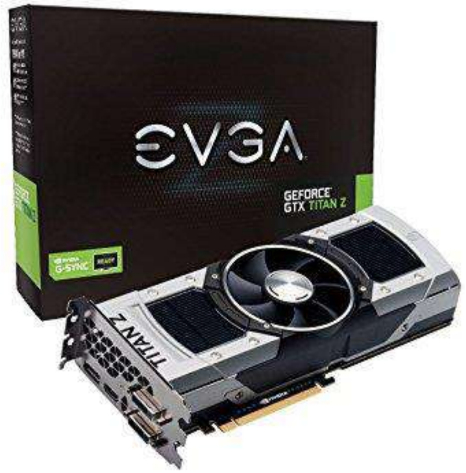
Advantages:

- *Amongst the very best performers | learners in the world*
- *Can be developed into deep learning algorithms*
- *Work with both regression and classification problems*

Disadvantages:

- *Requires large data set*
- *Black Box*
- *Prone to overfitting problem*
- *Need more power if more hidden layers*

Time to buy a
graphic card :D



ALGORITHMS ARE NOT CREATED EQUAL

- **Neural Nets** [Backpropagation]
- **SVM** [Kernel Tricks]
- **KNN** [Majority Vote]
- **Random Forest** [Randomness]
- **Decision Tree** [Recursive Partitioning]
- **Logistic Regression** [Sigmoid Function]
- **Linear Regression** [Least Squared]

R has powerful package written by Max Kuhn (10 years in development)

```
# CARET
# Classification And Regression Training
# 238 models can be trained using caret
install.packages("caret")
library(caret)
```

For more details, go to this website
http://topepo.github.io/caret/index.html