Introduction to Supervised Classification

Vijayabharathi (@ImtheVB)

Agenda

- What is Supervised Classification ?
- Classification algorithms
    - K- Nearest Neighbors
    - Support Vector Machine
    - Decision Tree
- How to measure the Model Performance ?
- Ensemble Methods for Classification
- Code Walkthrough

What is Supervised Classification ?

What is Machine Learning?

- Machine Learning is the science (and art) of programming computers so they can **learn from data.**

- Machine learning is the field of study that gives **computers the ability to learn without being explicitly programmed.** — *Arthur Samuel, 1959*

- Machine learning is a subset of AI which provides machines the ability to **learn automatically** and **improve from the experience.**

Supervised and Unsupervised Learning

- In **supervised** machine learning, machines **learn under guidance** by feeding **labelled data** and explicitly telling them the input and outputs.

- In **Unsupervised** machine learning, **machines are fed with unlabeled data** and it **has to find hidden patterns** in order to make prediction about the output.

- **Reinforcement** learning is a **hit and trail concept** where model will learn from the live experience by getting rewards and punishments (Intelligent systems)

Regression and Classification Problems

- Regression is a form of **predictive modelling technique** which investigates the relationship between independent and dependent variables to predict any unknown value.

Example: **Trend forecasting**

- Classification is the **process of categorizing** a given set of data into classes. It can be performed on both structured and un-structured data. Classes are often referred as target, label or categories

Example : **Disease or No disease**

What is Supervised Classification ?

- Supervised Classification = Supervised machine learning + Classification

- In supervised classification, the model is **trained with labelled data and expected to classify the unseen data** into one of the given categories based on experience / training .

- Example **Spam filter** is a machine learning program that can learn to flag spam after being given examples of spam emails that are flagged by users, and examples of regular non-spam emails.

Classification algorithms

Types of Classification Algorithms

A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.

Binary Classification: Classification task with two possible outcomes. Eg: Gender classification (Male / Female)

Multi-class classification: Classification with more than two classes. In multi class classification each sample is assigned to one and only one target label. Eg: An animal can be cat or dog but not both at the same time

Multi-label classification: Classification task where each sample is mapped to a set of target labels (more than one class). Eg: A news article can be about sports, a person, and location at the same time.

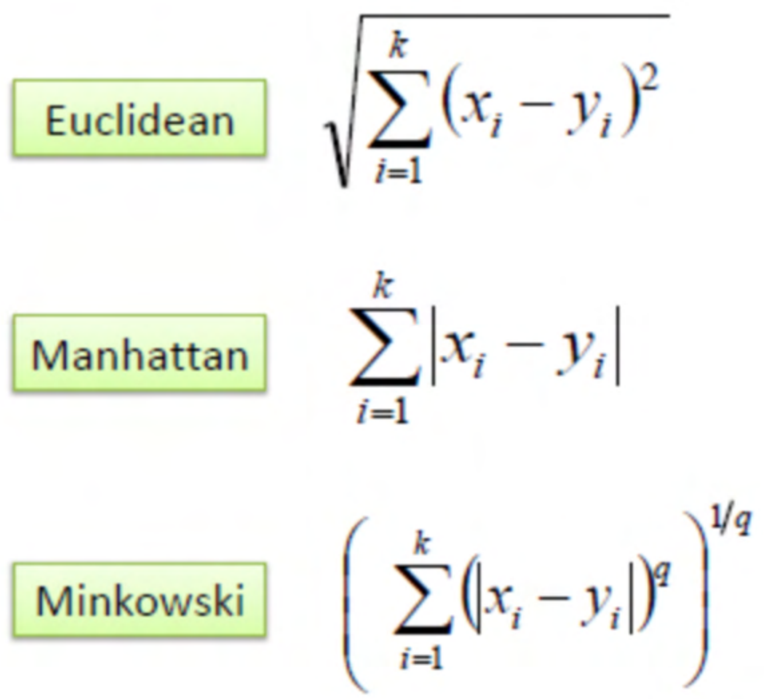
- Logistic Regression
- Naïve Bayes
- Stochastic Gradient Descent
- K-Nearest Neighbours
- Decision Tree
- Random Forest
- Support Vector Machine

K- Nearest Neighbors (KNN)

- The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

- KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label.

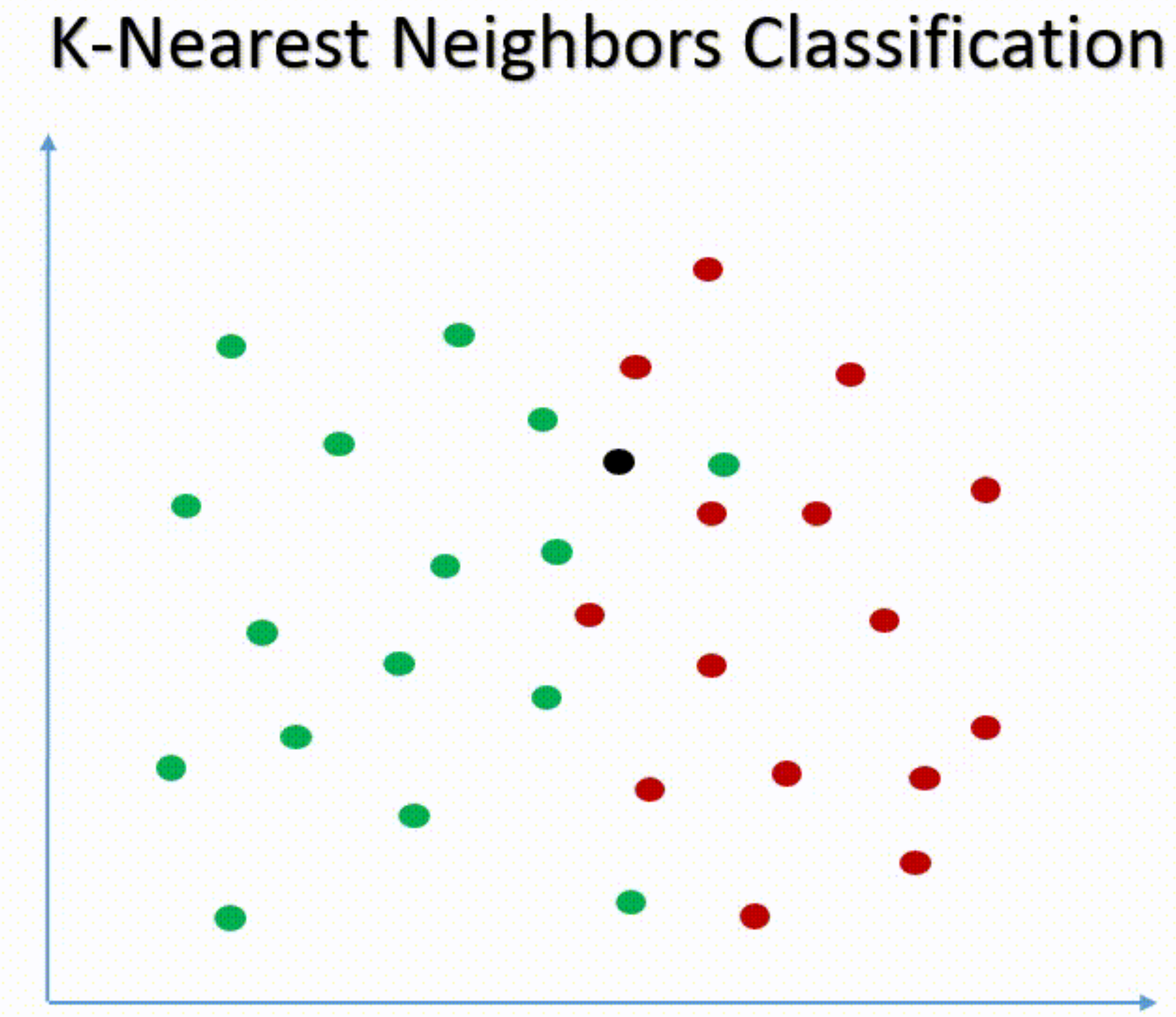
Note : Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value



Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$



K-Nearest Neighbors Classification

Image from machinelearningknowledge.ai

Support Vector Machine

- In the SVM algorithm, We plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes well.

- Support Vectors are simply the co-ordinates of individual observation. SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line).

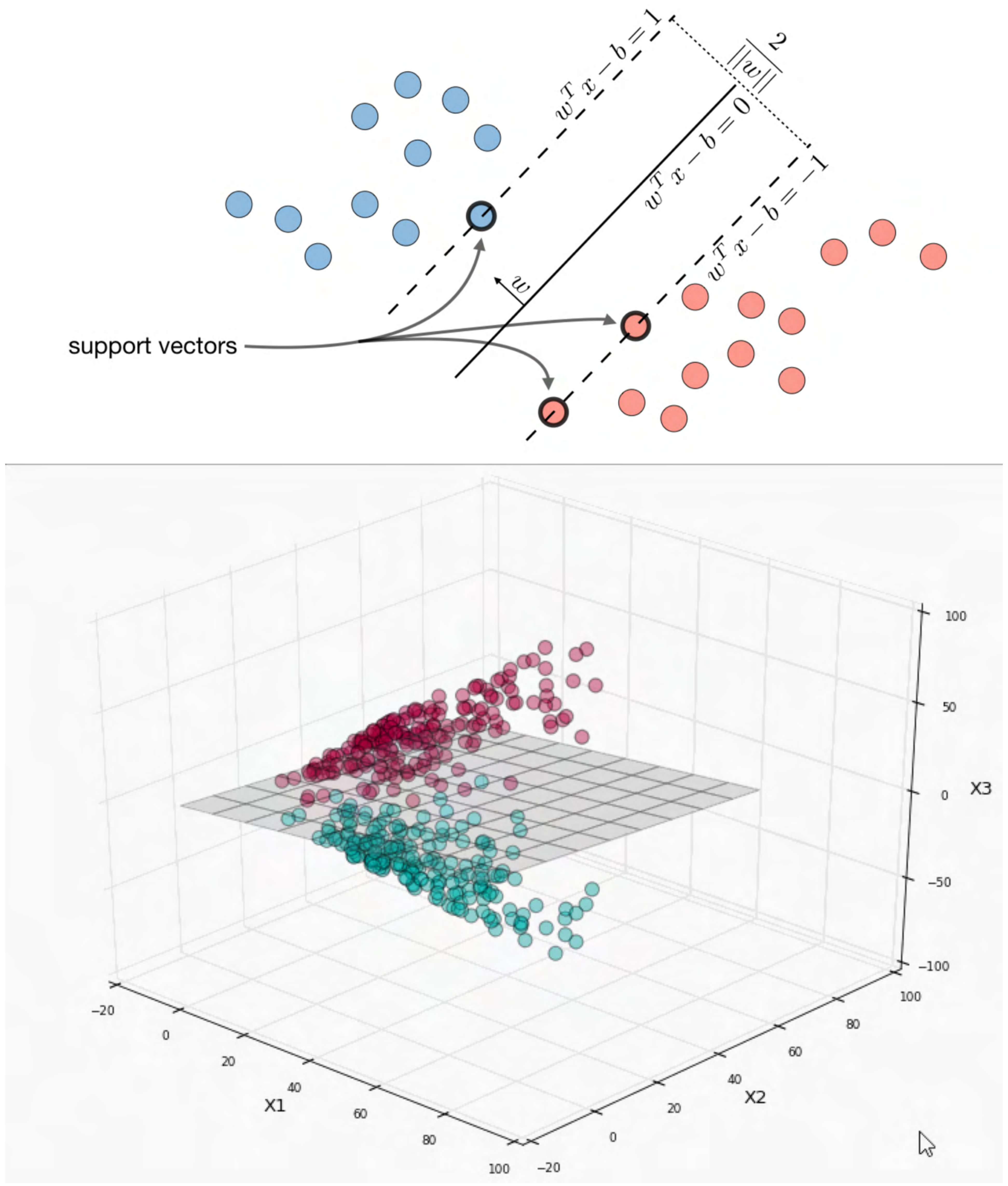


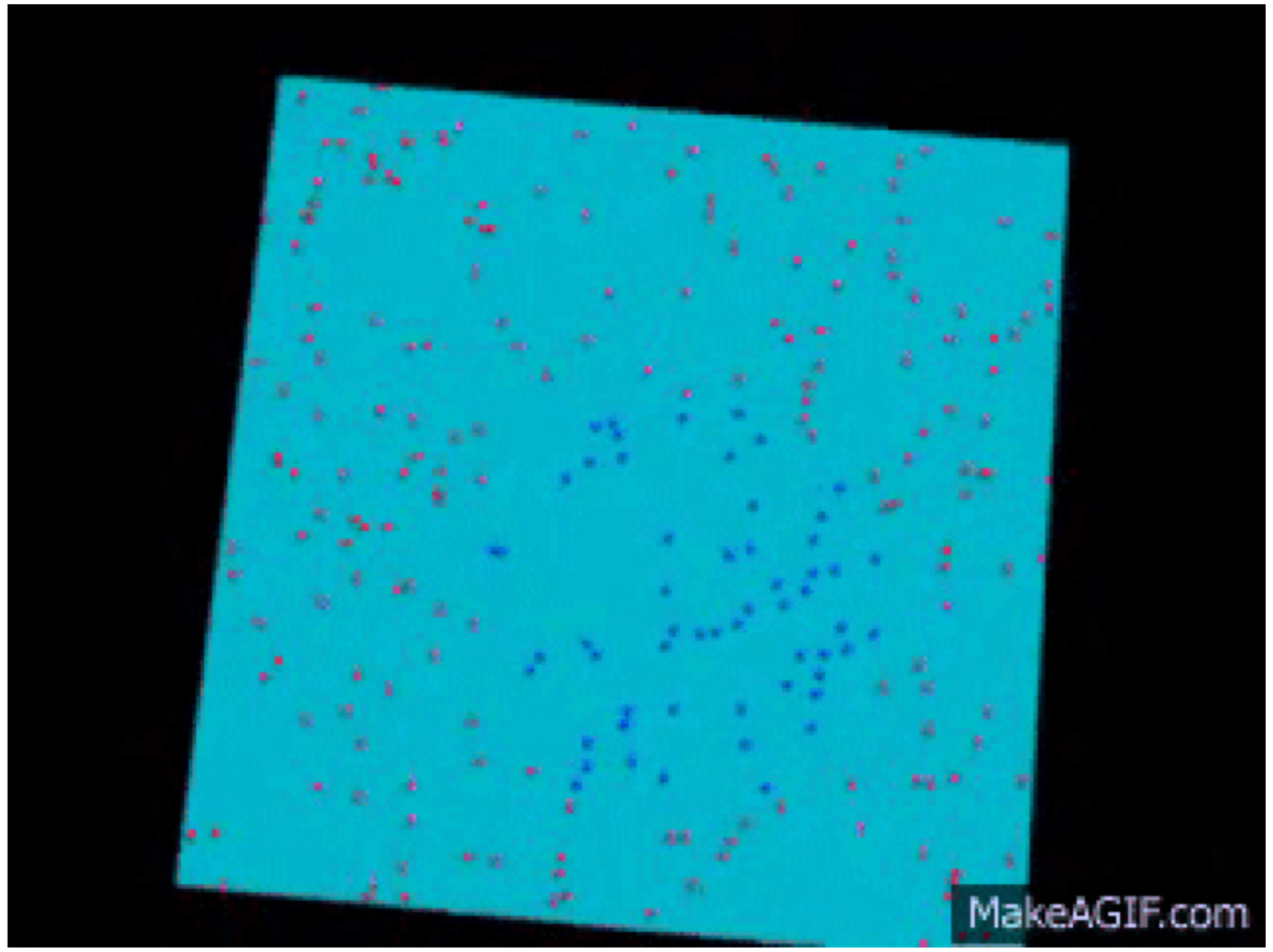
Image from blog.statsbot.co

SVM Kernel

- The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem.

- It does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs that are defined.

- Popular kernels are Linear, Rbf, Polynomial etc.



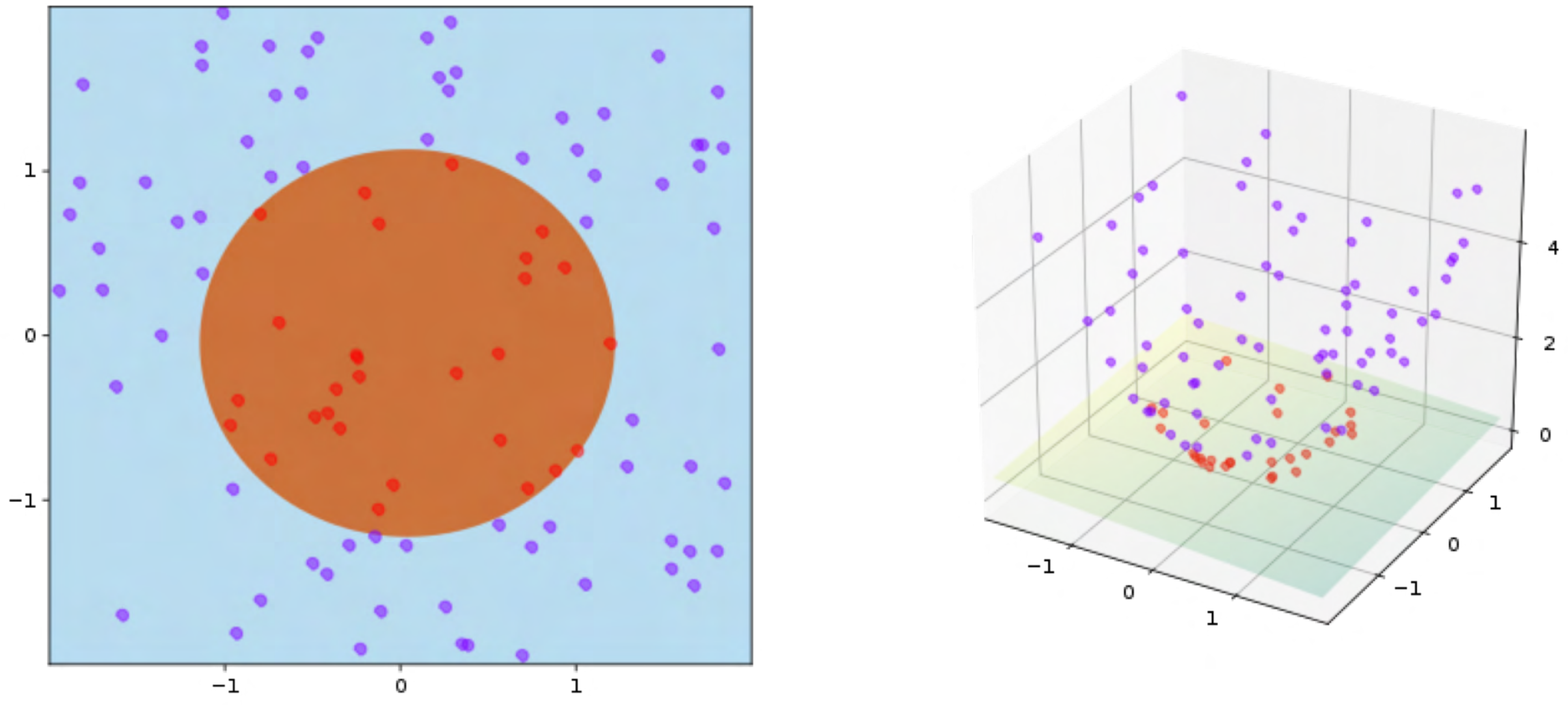


Image from wikipedia

Decision Tree

- In Decision tree, the data is continuously split according to a certain parameter. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

- Attribute selection measure is used select the best attribute for the root node and for sub-nodes.

- **Information Gain** (Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute)

- **Gini Index** (Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm. Low Gini index is preferred)

- **Pruning** (Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree)

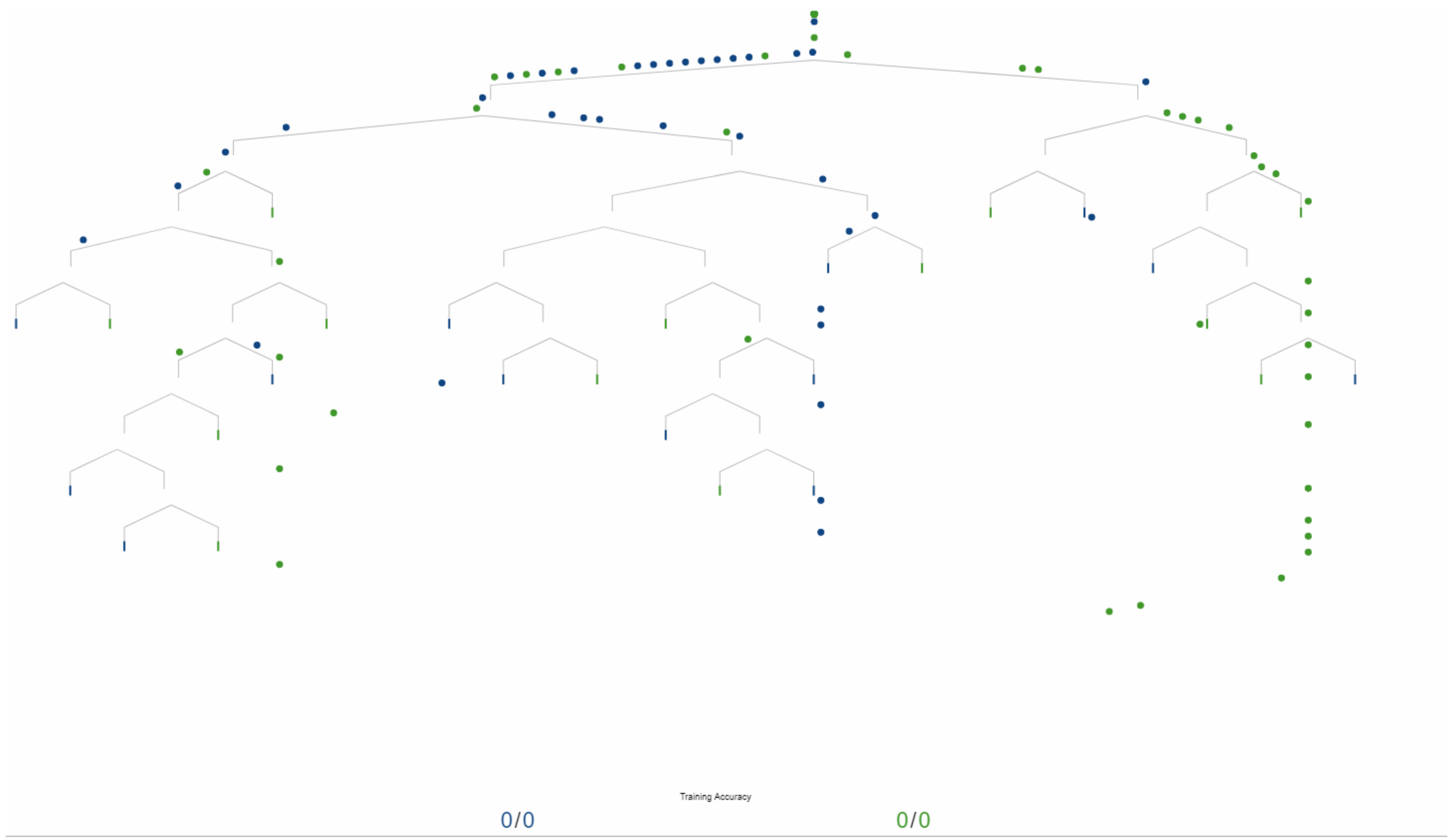


Image from r2d3.us

How to measure the Model Performance ?

Model metrics

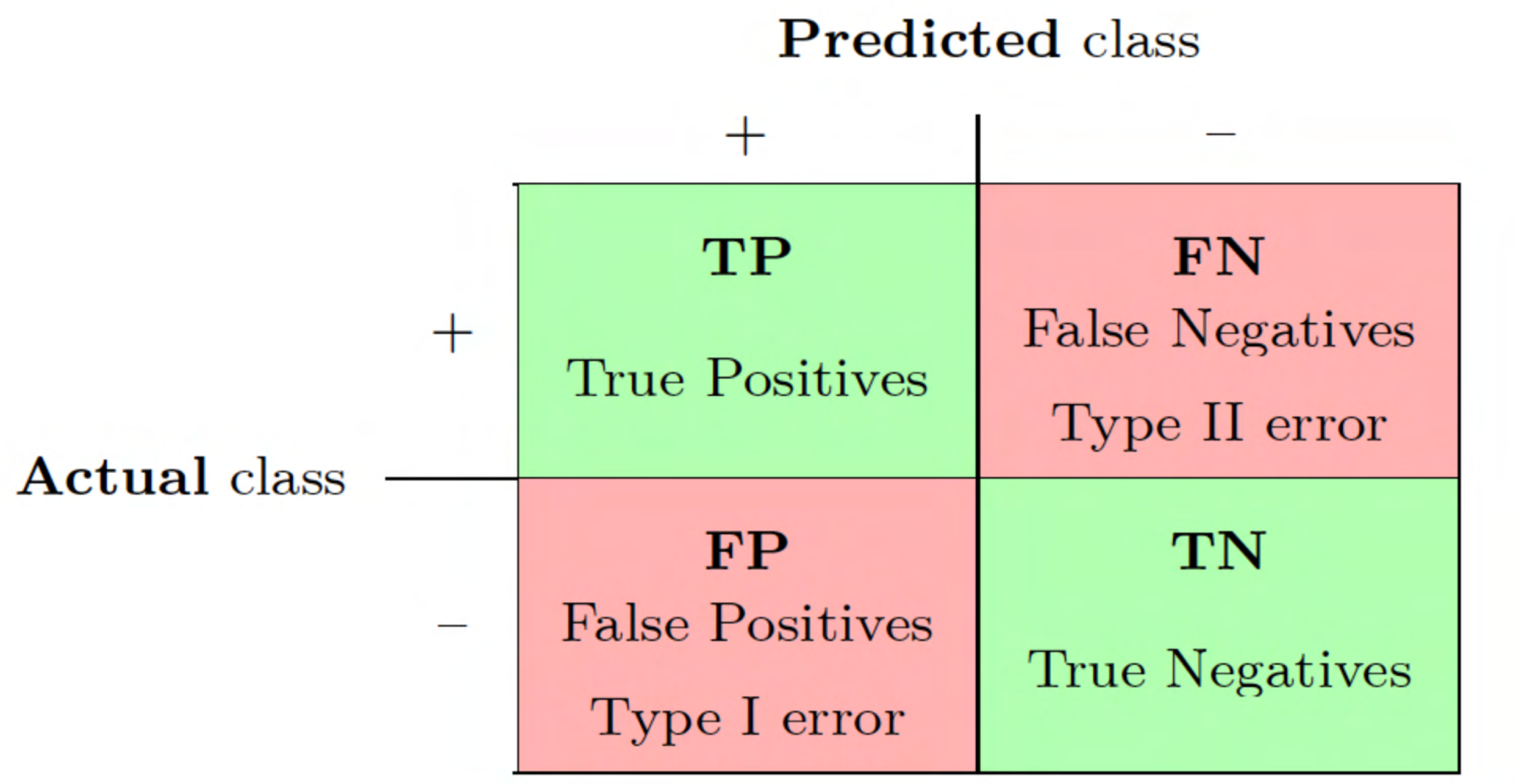
- In a context of a binary classification, metrics are important to track and assess the performance of the model.
- These will help to understand the exactness, completeness and correctness of the predictions to get more confidence on the model.

The following are the few majorly used Metrics for classification models

- Confusion matrix
- Accuracy, Precision, Recall, F1 score
- ROC and AUC
- CAP Curve

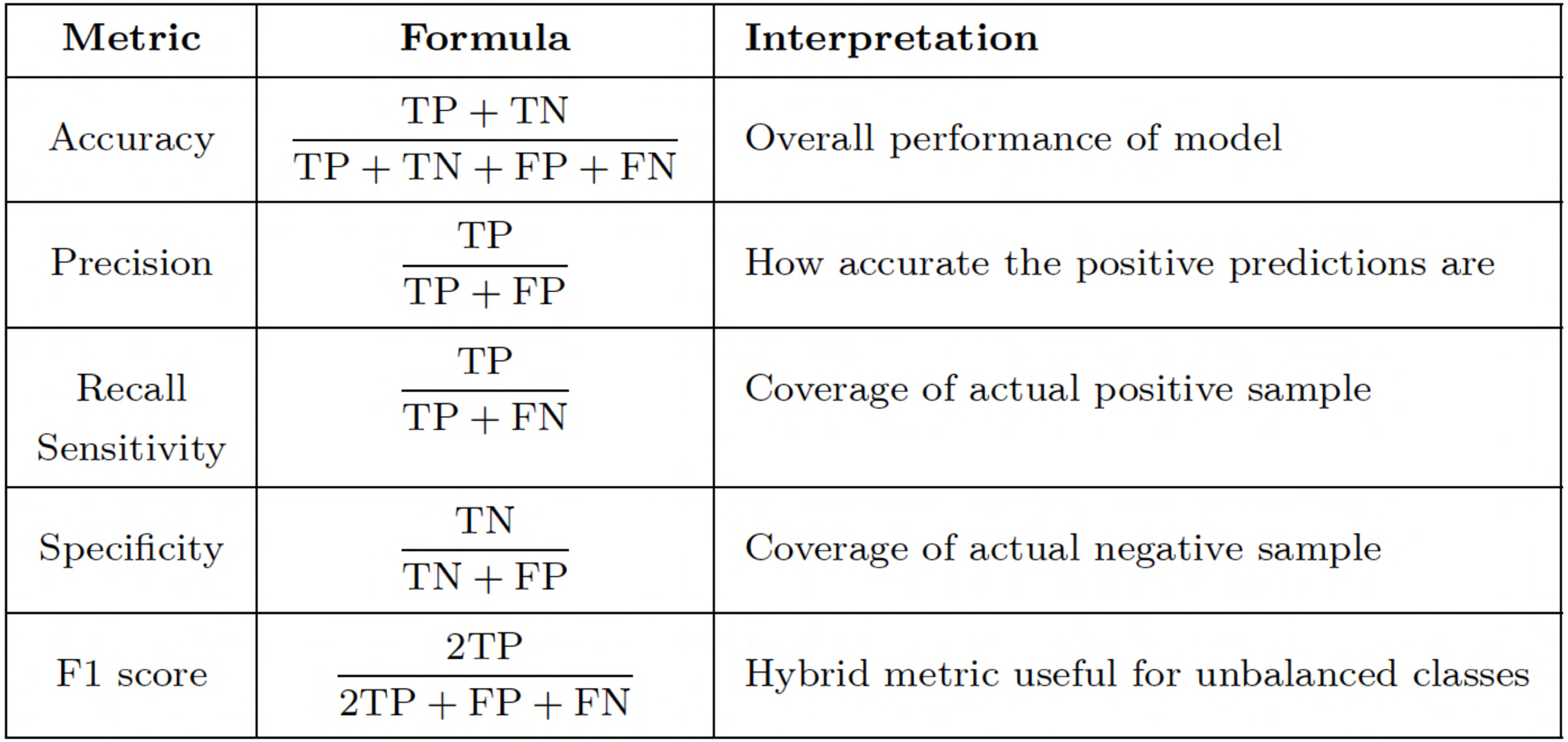
Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It is a table with four different combinations of predicted and actual values in the case for a binary classifier.



		Predicted class	
		+	−
Actual class	+	**TP** True Positives	**FN** False Negatives Type II error
	−	**FP** False Positives Type I error	**TN** True Negatives

Main metrics

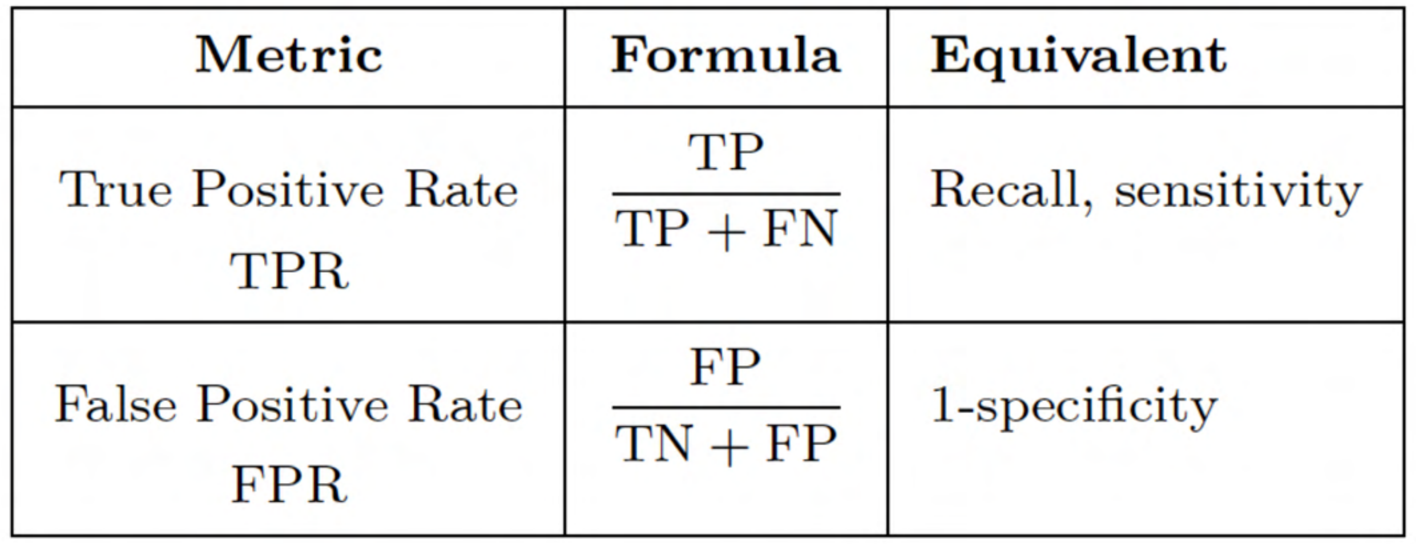


Metric	Formula	Interpretation
Accuracy	$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$	Overall performance of model
Precision	$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$	Coverage of actual positive sample
Specificity	$\frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$	Coverage of actual negative sample
F1 score	$\frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$	Hybrid metric useful for unbalanced classes

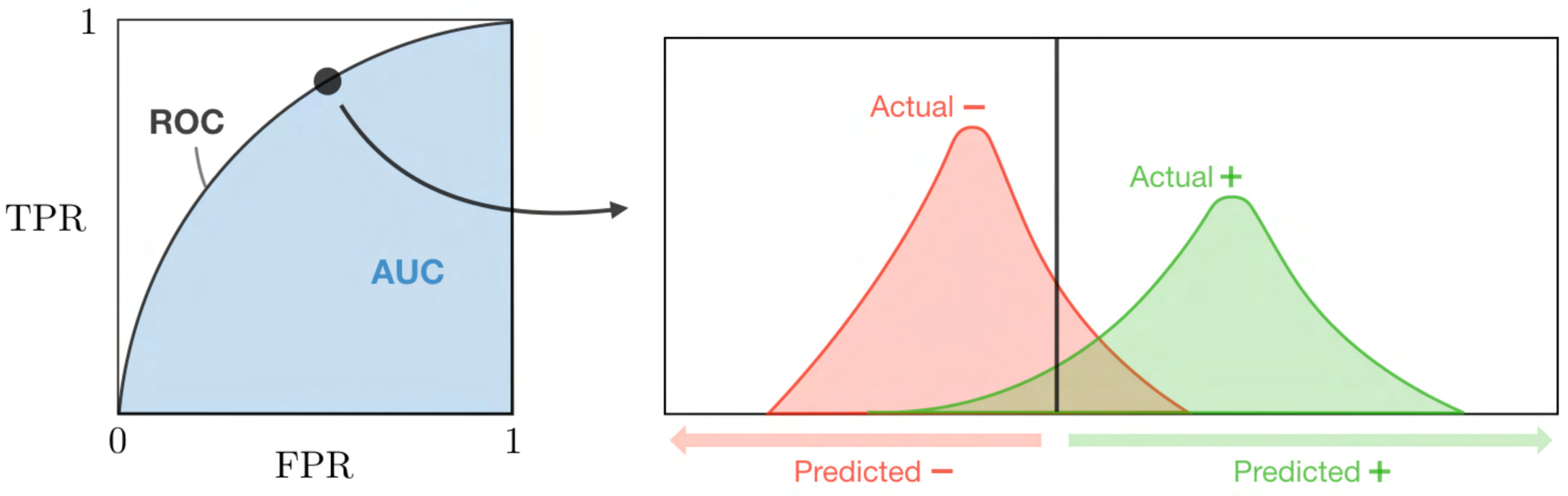
All the above values should be as high as possible

ROC and AUC

ROC (Receiver Operating Curve) tells us how well the model has accurately predicted. The ROC curve shows the sensitivity of the classifier. AUC means The area under the receiving operating curve. The better the AUC measure, the better the model.



Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{TP}{TP + FN}$	Recall, sensitivity
False Positive Rate FPR	$\frac{FP}{TN + FP}$	1-specificity



Bias and Variance

- Bias – The bias of a model is the difference between the expected prediction and the correct model that we try to predict for given data points.

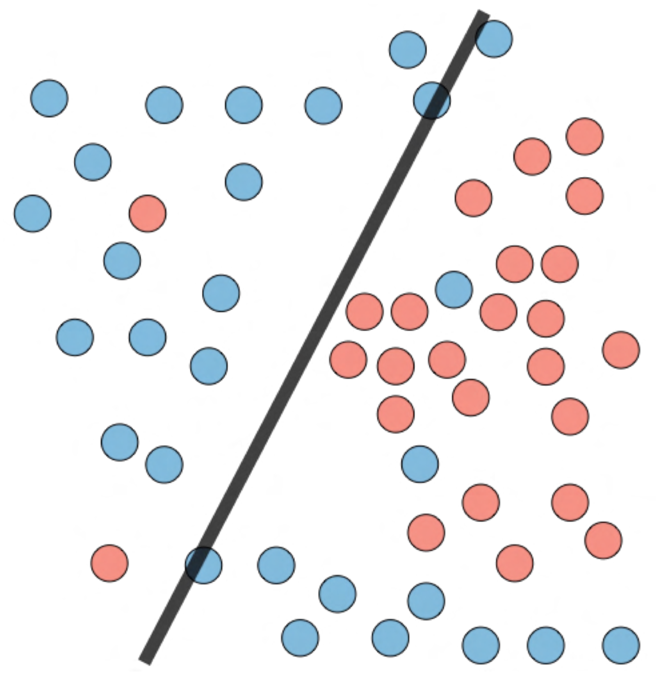
$$bias = \mathbb{E}[f'(x)] - f(x)$$

- Variance – The variance of a model is the **variability of the model prediction for given data** points.

$$variance = \mathbb{E}\left[\left(f'(x) - \mathbb{E}[f'(x)]\right)^2\right]$$

- Bias/variance trade-off – The simpler the model, the higher the bias, and the more complex the model, the higher the variance.

Effects of Bias and Variance

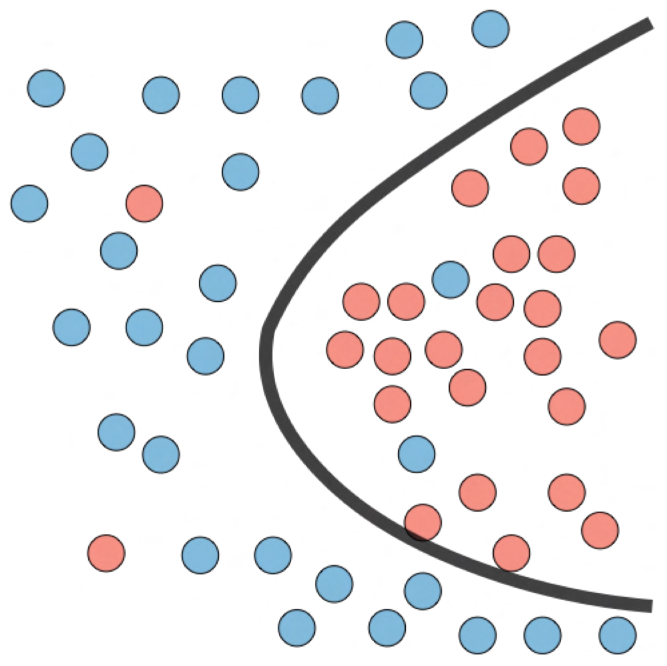


Underfitting

- High training error
- Training error close to Test error
- High bias

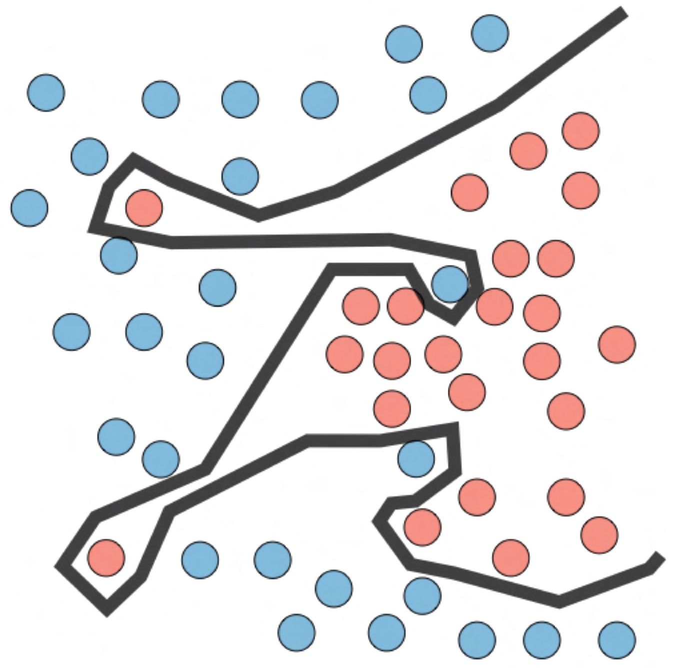
To rectify Underfitting

- Complexify model
- Add more features
- Train longer



Just right model

Training error slightly lower than Test error



Overfitting

- Low training error
- Training error **much lower** than Test error
- High variance

To rectify Overfitting

- Regularize the model
- Get more data

Ensemble Methods for Classification

Ensemble Methods

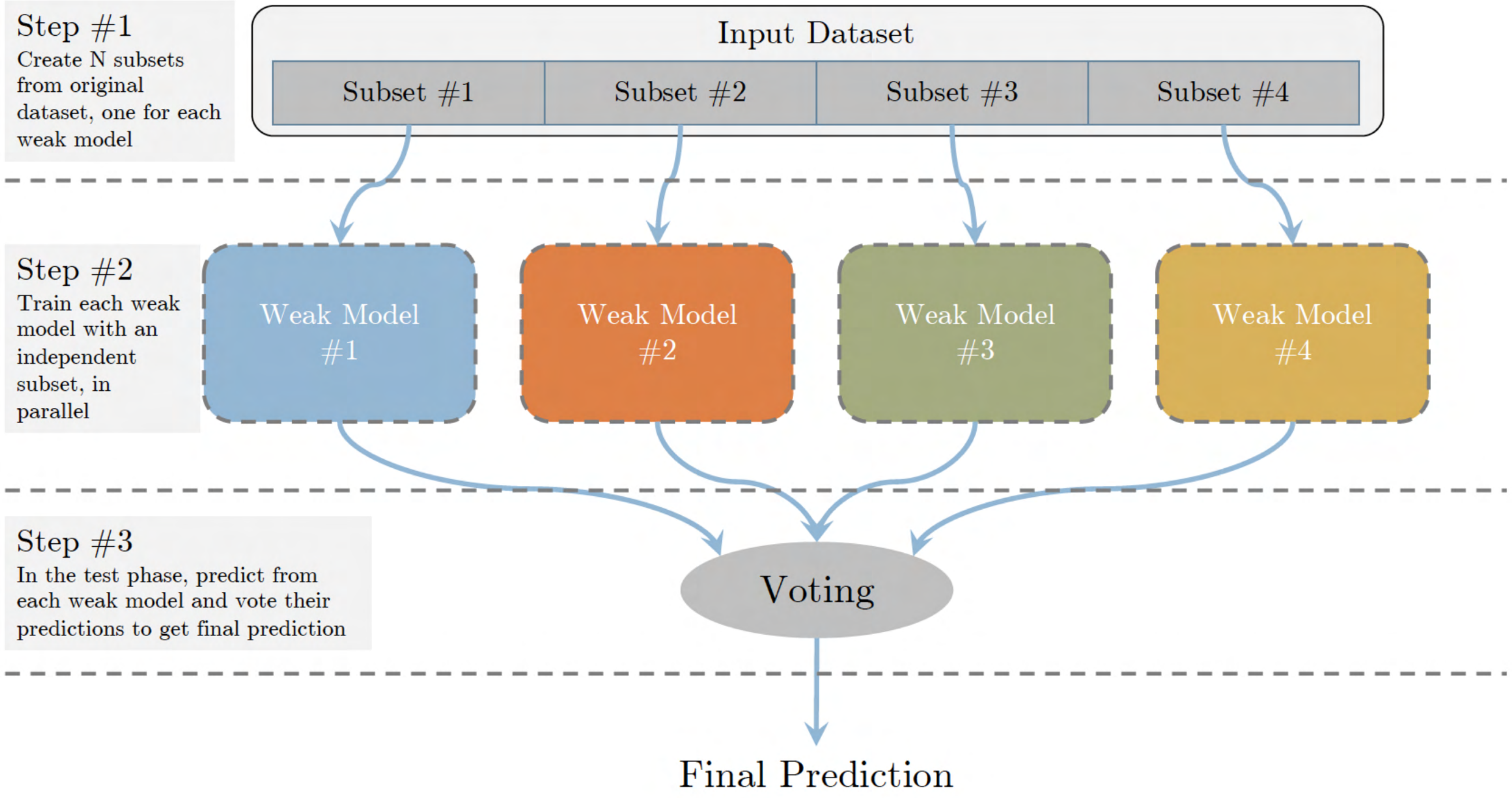
- Ensemble method is a way of combining several supervised learning models that are individually trained and the results merged in various ways to achieve the final prediction. This result has higher predictive power than individual algorithms.
- Combine multiple weak models/learners into one predictive model to reduce bias, variance and/or improve accuracy.

Types of Ensemble techniques

- Bagging (Random Forest)
- Boosting (AdaBoost, Gradient Boosting)
- Stacking

Bagging

- Bagging: Trains N different weak models (usually of same types – homogenous) with N non-overlapping subset of the input dataset in parallel. The label with the greatest number of predictions is selected as the prediction. Bagging methods reduces variance of the prediction



Step #1
Create N subsets from original dataset, one for each weak model

Step #2
Train each weak model with an independent subset, in parallel

Step #3
In the test phase, predict from each weak model and vote their predictions to get final prediction

Bagging

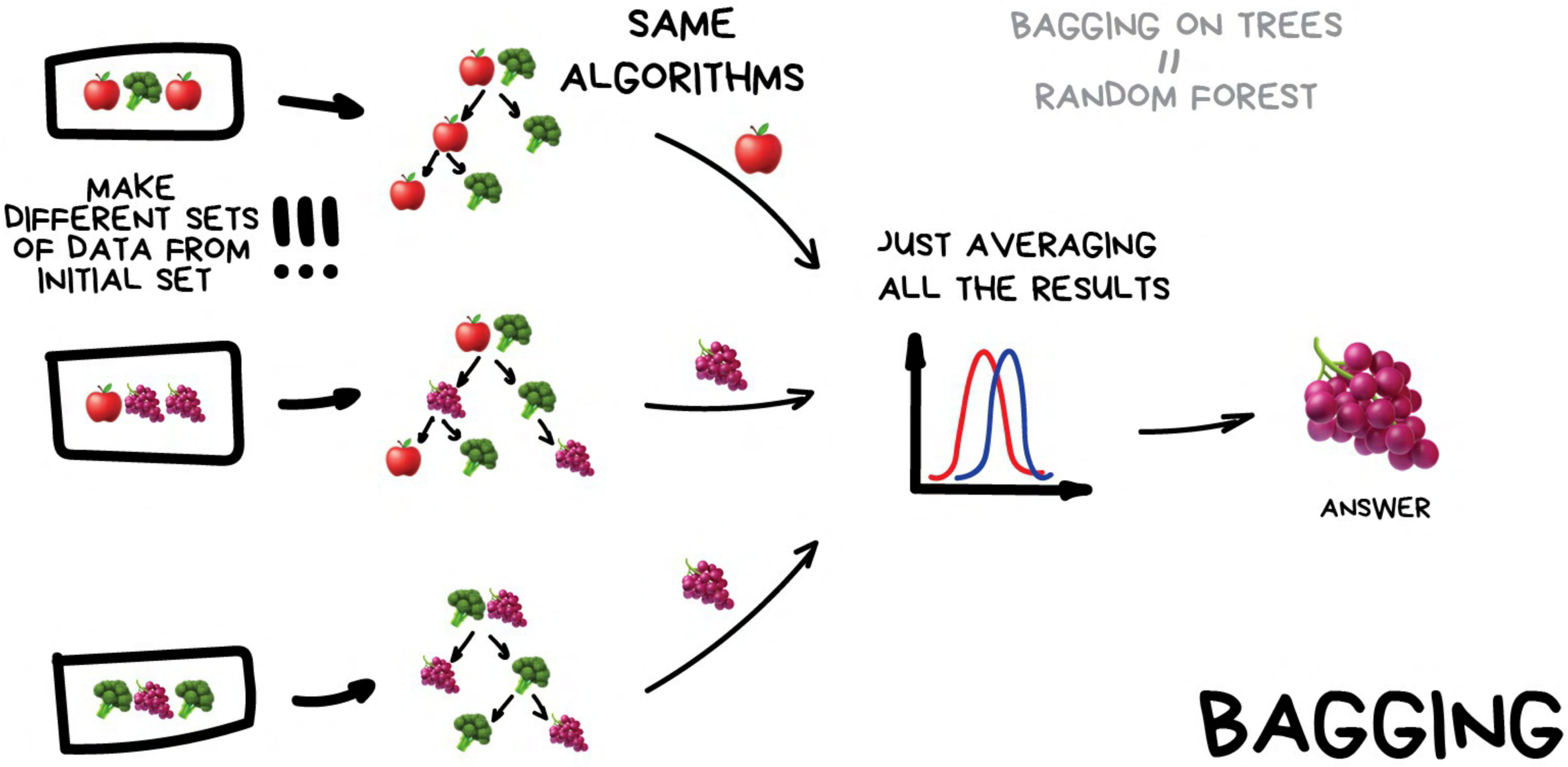
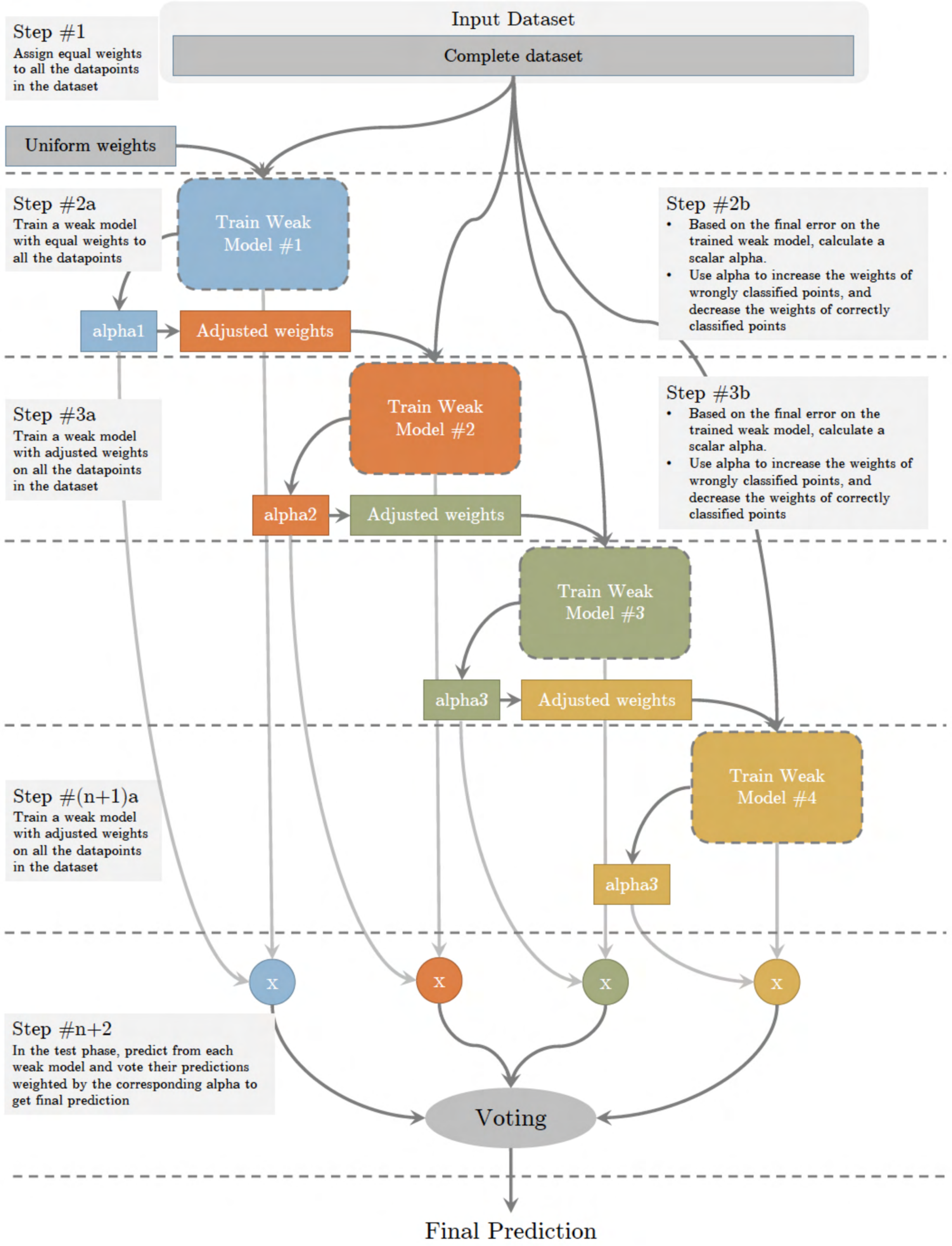

SAME ALGORITHMS
BAGGING ON TREES
//
RANDOM FOREST
MAKE DIFFERENT SETS OF DATA FROM INITIAL SET !!!
JUST AVERAGING ALL THE RESULTS
ANSWER
BAGGING

Image by @drangshu

Boosting

- Trains N different weak models (usually of same types – homogenous) with the complete dataset in a sequential order.
- The data points wrongly classified with previous weak model is provided more weights to that they can be classified by the next weak leaner properly.
- In the test phase, each model is evaluated and based on the test error of each weak model, the prediction is weighted for voting. Boosting methods decreases the bias of the prediction.



Step #1
Assign equal weights to all the datapoints in the dataset

Step #2a
Train a weak model with equal weights to all the datapoints

Step #2b
- Based on the final error on the trained weak model, calculate a scalar alpha.
- Use alpha to increase the weights of wrongly classified points, and decrease the weights of correctly classified points

Step #3a
Train a weak model with adjusted weights on all the datapoints in the dataset

Step #3b
- Based on the final error on the trained weak model, calculate a scalar alpha.
- Use alpha to increase the weights of wrongly classified points, and decrease the weights of correctly classified points

Step #(n+1)a
Train a weak model with adjusted weights on all the datapoints in the dataset

Step #n+2
In the test phase, predict from each weak model and vote their predictions weighted by the corresponding alpha to get final prediction

Boosting

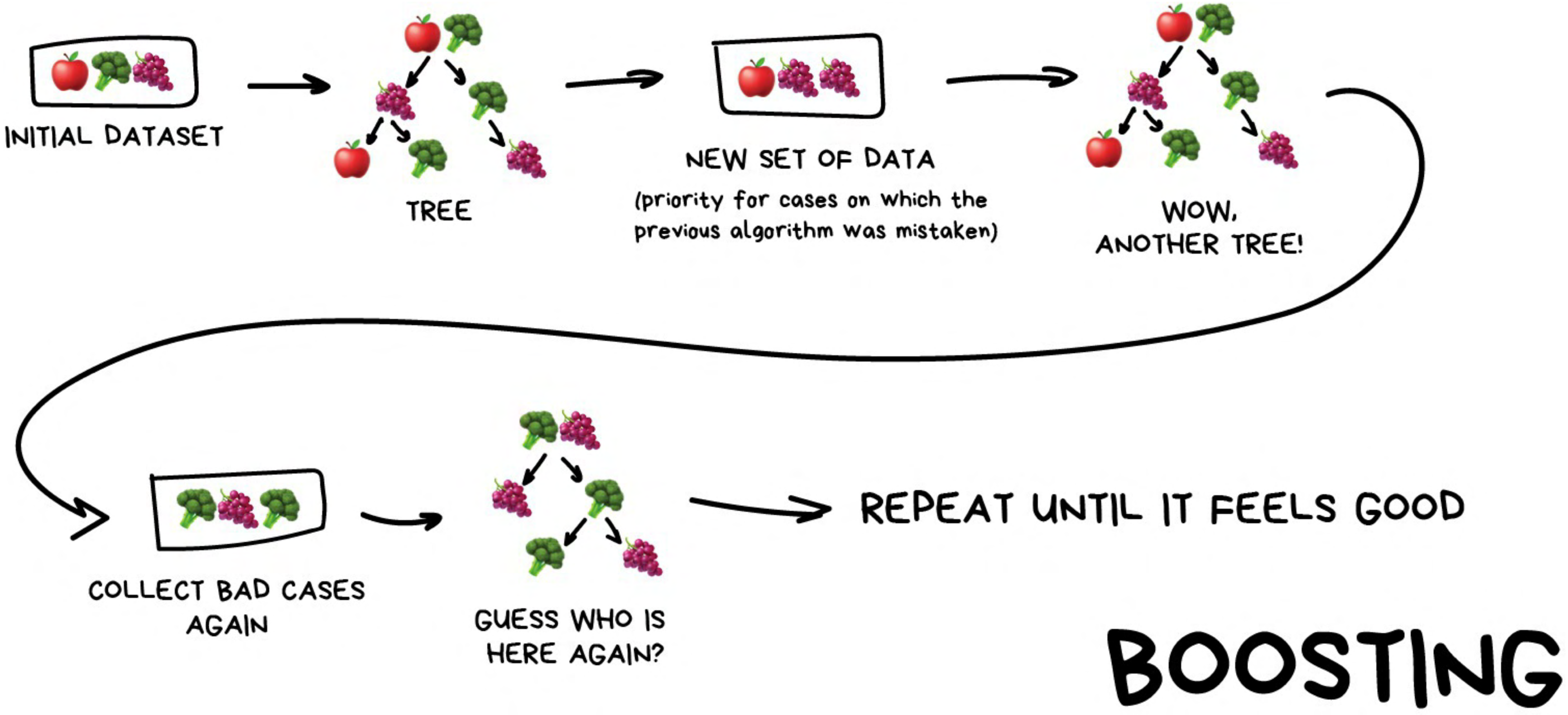
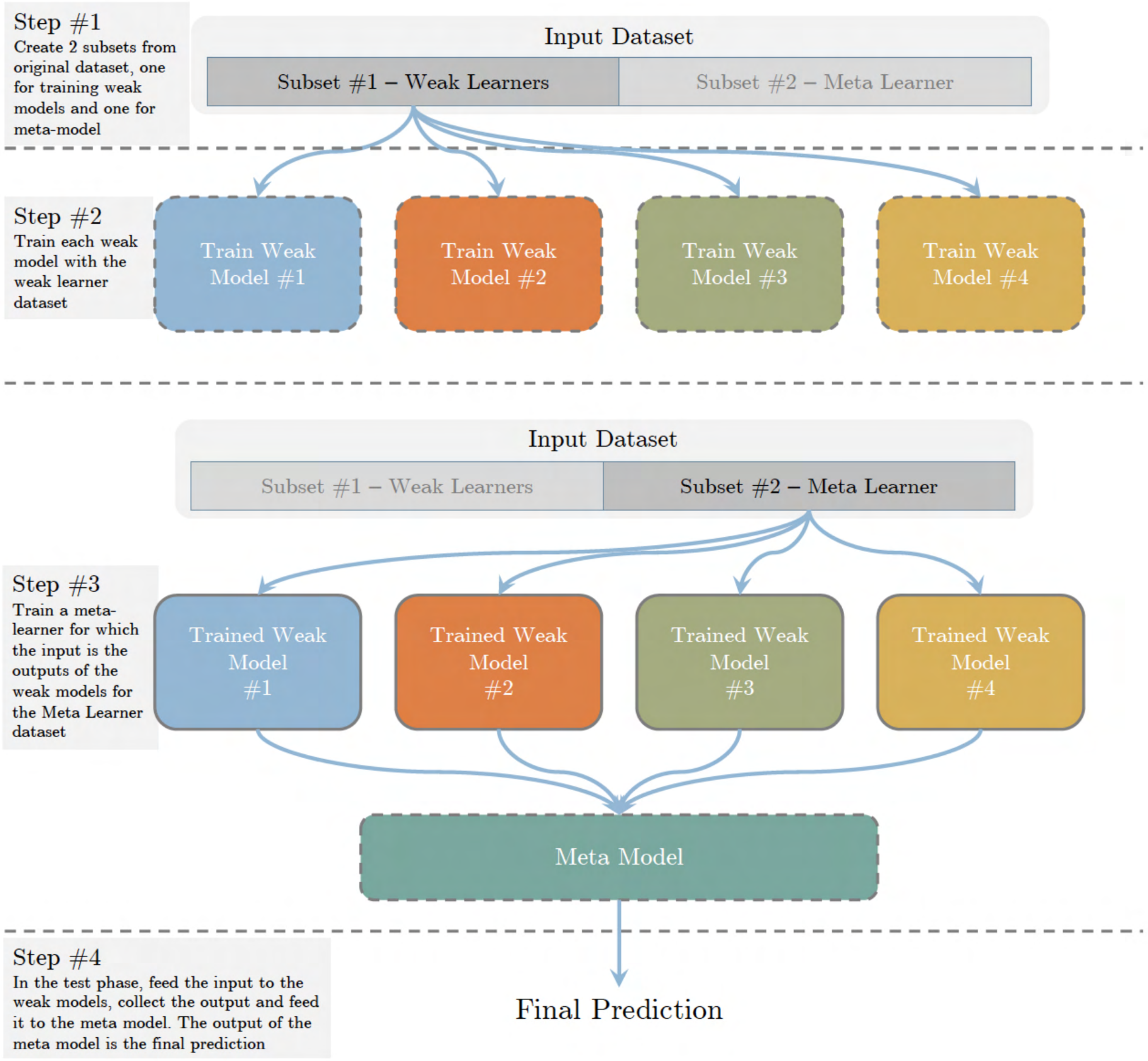


Image by @drangshu

Stacking

- Stacking: Trains N different weak models (usually of different types – heterogenous) with one of the two subsets of the dataset in parallel.
- Once the weak learners are trained, they are used to train a meta learner to combine their predictions and carry out final prediction using the other subset.
- In test phase, each model predicts its label, these set of labels are fed to the meta learner which generates the final prediction.



Step #1
Create 2 subsets from original dataset, one for training weak models and one for meta-model

Step #2
Train each weak model with the weak learner dataset

Step #3
Train a meta-learner for which the input is the outputs of the weak models for the Meta Learner dataset

Step #4
In the test phase, feed the input to the weak models, collect the output and feed it to the meta model. The output of the meta model is the final prediction

Stacking

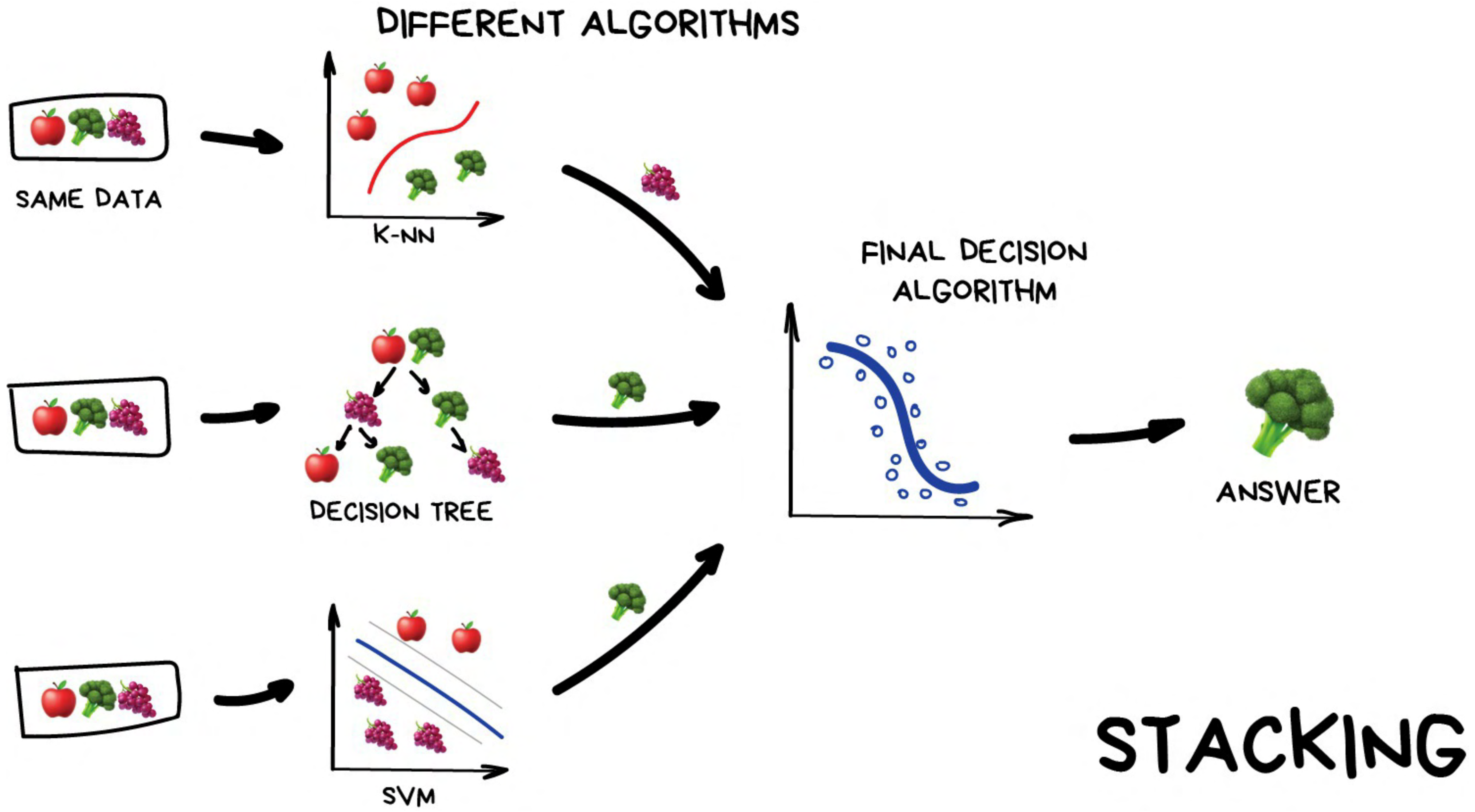


Image by @drangshu

Code walkthrough - Demo

Thank you...!