Supervised Learning - Classification

15-17 November 2021

SUPERVISED MACHINE LEARNING ALGORITHMS[1]

- Are applied when given data are ***classified or labeled***.
- Train models with labelled data then predict the output (known output values)
- Learn the mapping function from the input ***x*** to the output ***y:*** ***y = h(x)***

Goal: approximate the mapping function so well that it can be used to predict the output ***y*** of new input data ***x***

- Algorithms learn to make predictions on the training data, while supervised by labels
- Learning stops when achieving an acceptable level of performance

SUPERVISED MACHINE LEARNING ALGORITHMS[3]

Let's assume our simple predictor has this form: $h(x) = \theta_0 + \theta_1 x$

- Goal: find the values of θ_0 and θ_1 to make our predictor work as well as possible.

Optimizing the predictor ***h(x)*** is done using training examples.

- For each training example, we have an input value ***x_train***, for which a corresponding output, ***y***, is known in advance.
- For each example, we find the difference between the known, correct value ***y***, and our predicted value ***h(x_train)***.
- With enough training examples, these differences give us a useful way to measure the "wrongness" of ***h(x)***.
- We can then tweak ***h(x)*** by tweaking the values of θ_0 and θ_1 to make it "less wrong".
- This process is repeated over and over until the system has converged on the best values for θ_0 and θ_1
- In this way, the predictor becomes trained, and is ready to do some real-world predicting.

SUPERVISED MACHINE LEARNING ALGORITHMS[4]

Classification: identify category of new observations on the basis of training data

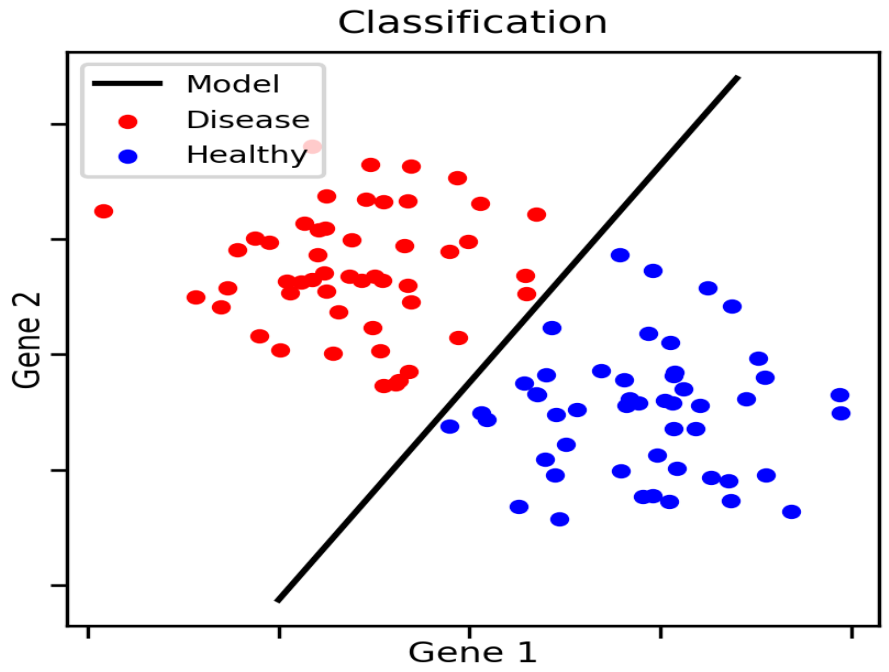
- e.g. binary classifier: is this tumor cancerous?, is this email a spam?; multi-class classifier: classification of types of music, virus variants

Regression: model the relationship between a dependent (target) and independent (predictor) variables

- e.g. salary of employees ~ year of experience, gene expression ~ genetic variants (eQTL)

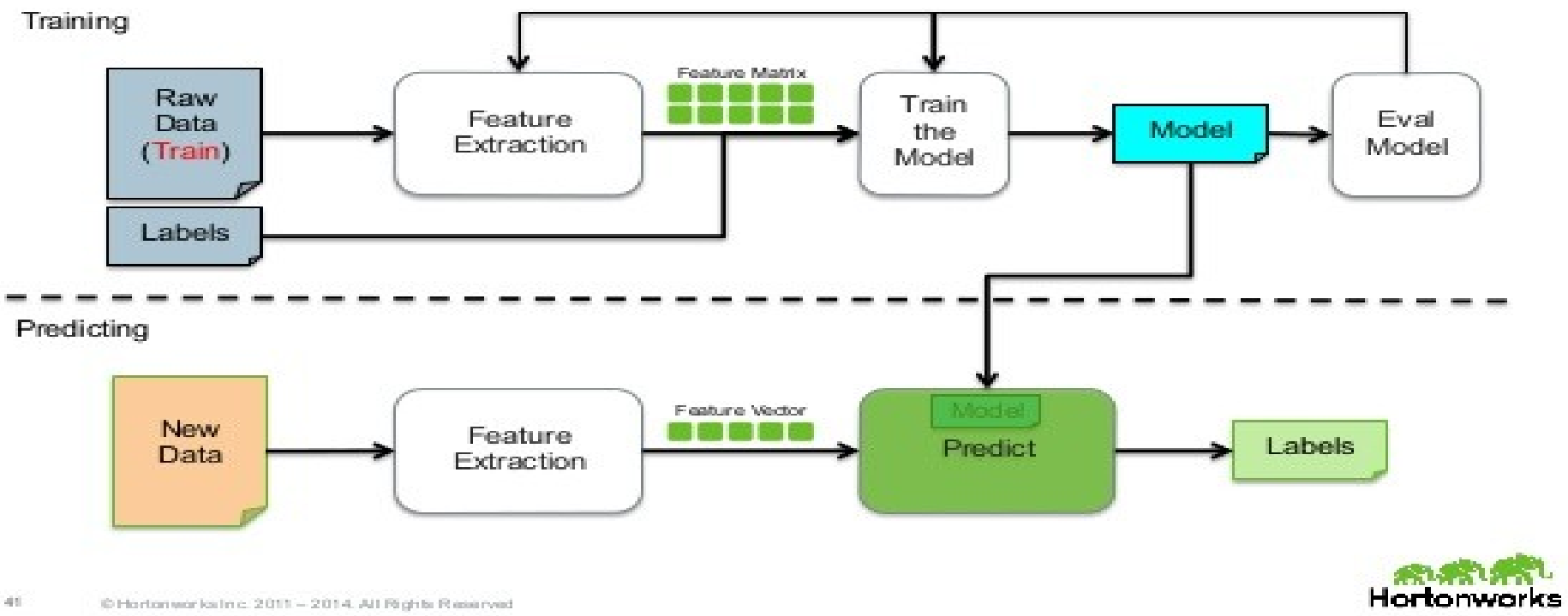
SUPERVISED MACHINE LEARNING ALGORITHMS[5]

CLASSIFICATION VS REGRESSION

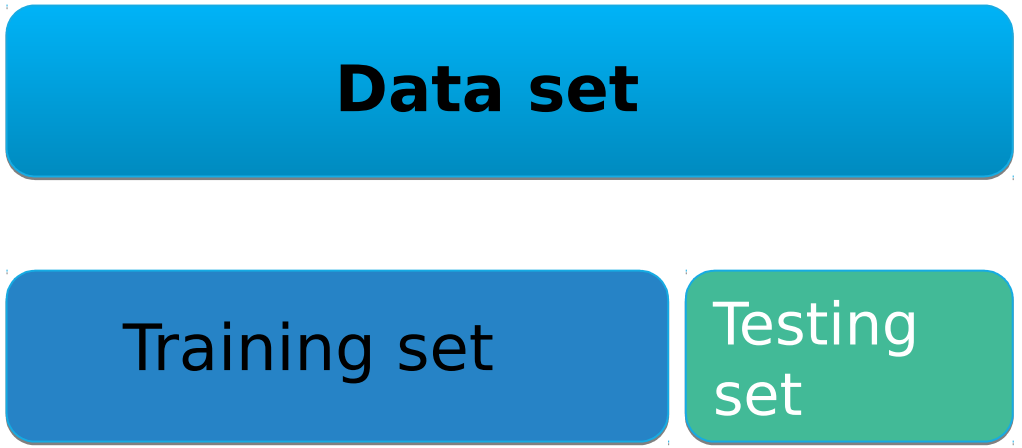


https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/

SUPERVISED MACHINE LEARNING ALGORITHMS[2]



TRAINING SET AND TEST SET

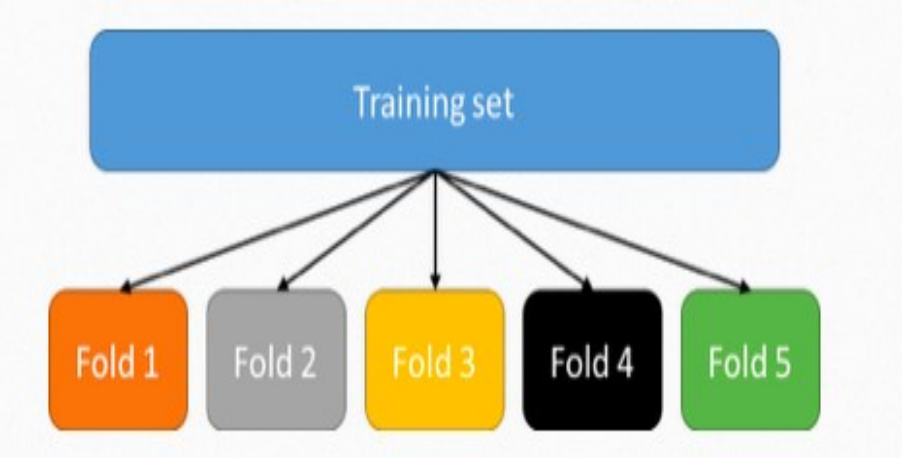


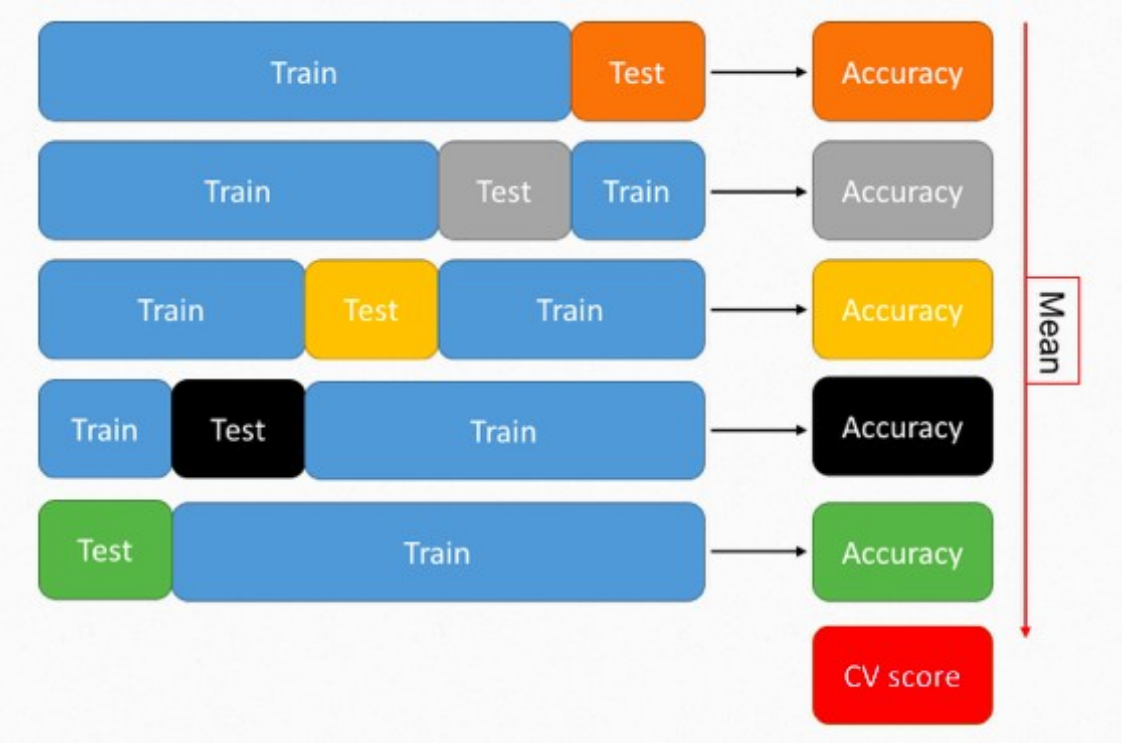
Used to train the algorithm

Estimate the accuracy of the model

Split the dataset randomly!
Use cross-validation
Underfitting and over fitting problems

K-FOLD CROSS VALIDATION





https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/#type-of-learning-problems

SUPERVISED LEARNING - DATASETS

Training dataset
A subset of the dataset provided to the algorithm for learning

Validation dataset
A subset used to tune the trained model parameters

Test dataset
A dataset used only to assess the performance of a fully-specified model (classifier/regressor)

VALIDATION OF SUPERVISED ML ALGORITHMS RESULTS

To test the performance of the learning system:

- The trained model can be tested with objects with known labels (and were excluded from the training set because they were intended to be used for this purpose).
- Based on the results on the test data, the performance of the learning system can be assessed.

EXAMPLES OF SUPERVISED LEARNING ALGORITHMS

DECISION TREES (SUPERVISED)

A decision tree is a tree-like graph with

- Nodes: places for an attribute
- Edges: rules
- Leaves: actual outputs or class labels

Single trees are used very rarely, but in composition with many others they build very efficient algorithms such as Random Forest or Gradient Tree Boosting.

Used for both classification and regression tasks.

DECISION TREES (SUPERVISED)

Advantages:

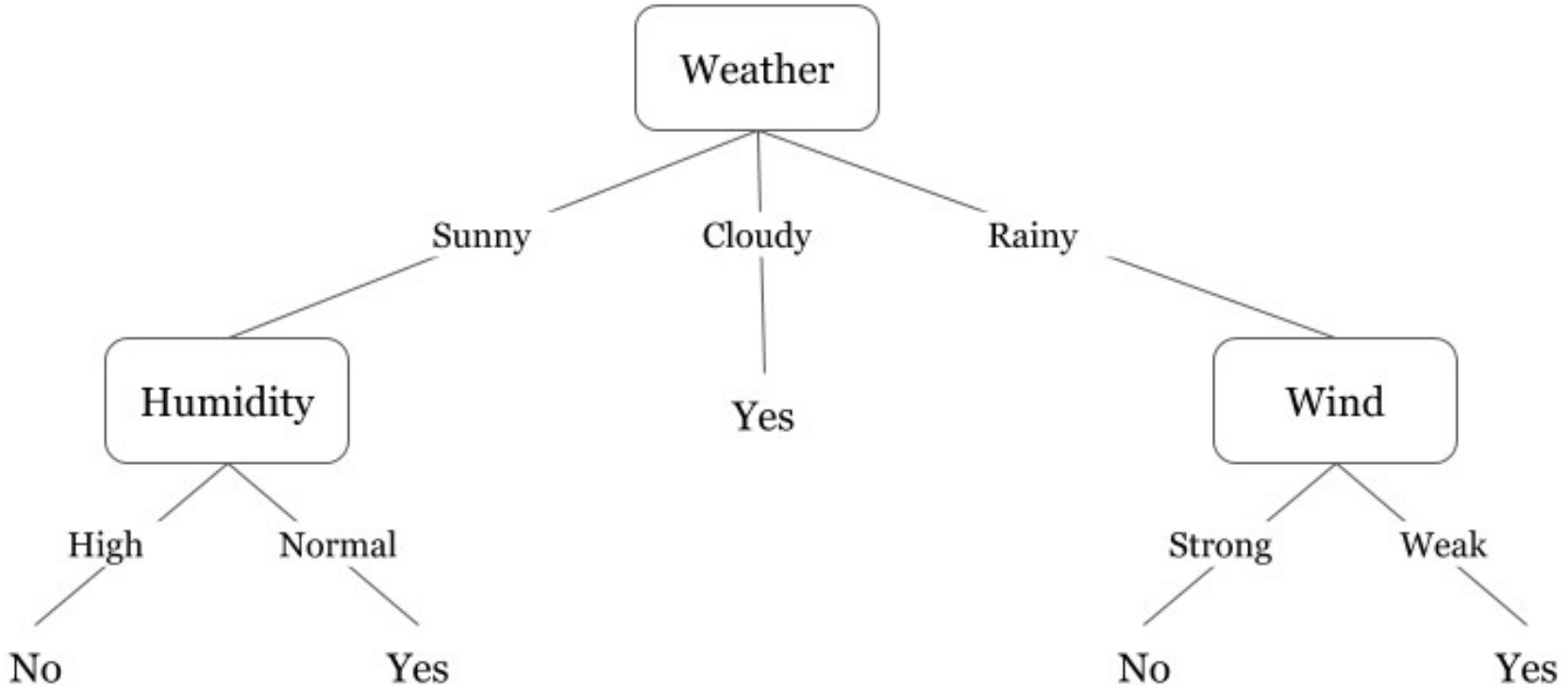
- Simple linear decision surface for non-linear decision making
- Easily handle feature interactions
- Non-parametric
- Dealing with outliers
- Solve **both regression and classification** problems

Disadvantages:

- Often the tree needs to be rebuilt when new examples come on.
- Easily overfit, but ensemble methods like random forests (or boosted trees) take care of this problem.
- Take a lot of memory (the more features you have, the deeper and larger your decision tree is likely to be)

E.g. Classification of genomic islands using decision trees and ensemble algorithms

DECISION TREES (SUPERVISED)



Classification metrics

WHY THE NEED TO EVALUATE?

- Multiple methods are available to classify or predict
- For each method, multiple choices are available for settings
- To choose best model, need to assess each model’s performance

MISCLASSIFICATION ERROR

- **Error** = classifying a record as belonging to one class when it belongs to another class.

- **Error rate** = percent of misclassified records out of the total records in the validation data

DIFFERENT SCORING METRICS

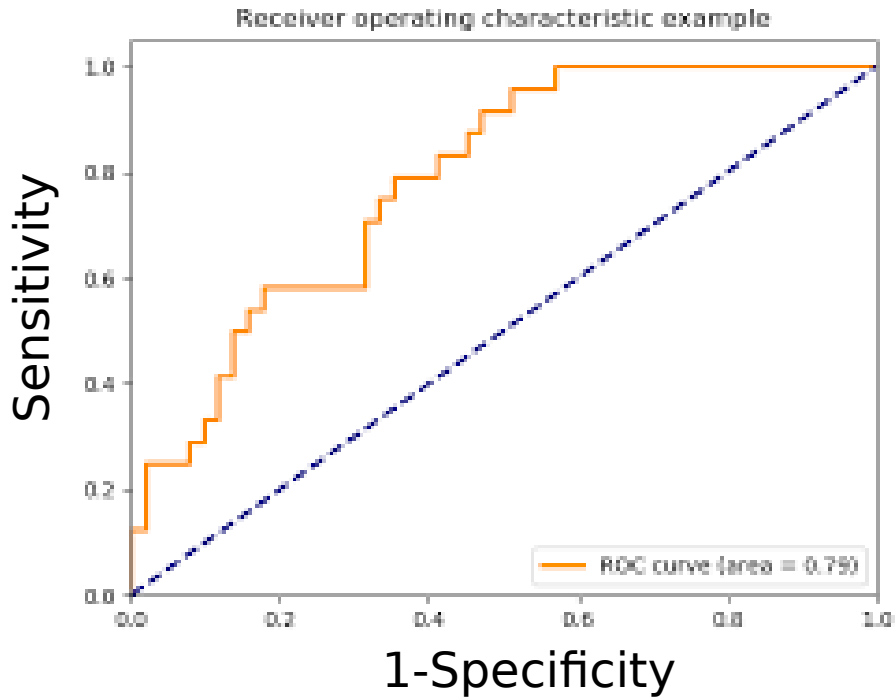
1. Confusion Matrix
   - True positives
   - False negatives
   - False positives
   - True negatives
2. Sensitivity and Specificty
3. Precision and Recall
4. F-measure
5. Overall accuracy and Cohen’s kappa

MAIN DEFINITIONS

- Confusion matri

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

- Precision $\frac{tp}{tp+fp}$
- Specificit $\frac{TN}{FP+TN}$
- Recall / Sensitivity $\frac{tp}{tp+fn}$



- Receiver Operating Characteristic (ROC) and AUC curves

https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.htm

F-MEASURE

$$\text{F-measure} = 2 * \frac{precision * recall}{precision + recall}$$

Harmonic mean of precision and recall

Are ALL and ONLY positive class events found by the model?

OVERALL ACCURACY

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Target class distribution must be balanced!

Probability of classifying a positive OR negative class event correctly.

WHY DIFFERENT METRICS?

1. What is your objectives?

2. What is the target class distribution?

3. Is the target binomial or multinomial?

BACK ON TRACK - EXAMPLES OF SUPERVISED LEARNING ALGORITHMS

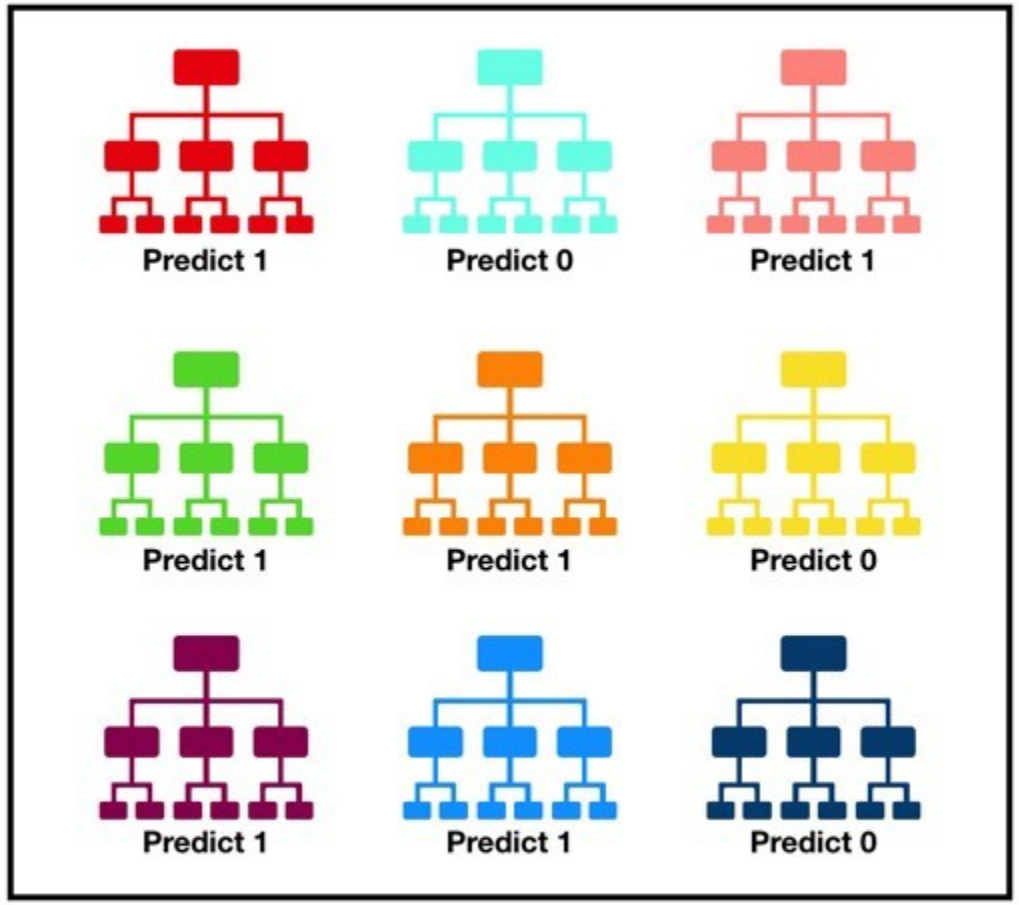
RANDOM FOREST (SUPERVISED)

Random forest: multiple individual decision trees operating as an ensemble.

One class prediction from each individual tree

=> model's prediction = class with most votes

=> more accurate and stable prediction



RANDOM FOREST (SUPERVISED)

Advantages:

- Solve **both regression and classification** problems with large data sets.
- Help identify most significant variables from thousands of input variables.
- Highly scalable to any number of dimensions with generally quite acceptable performances.

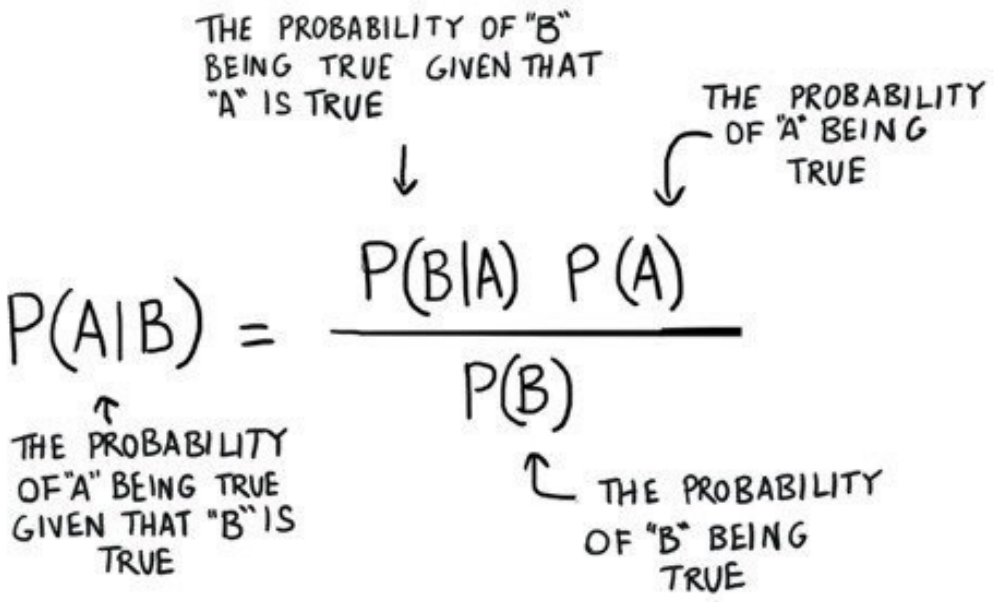
Disadvantages:

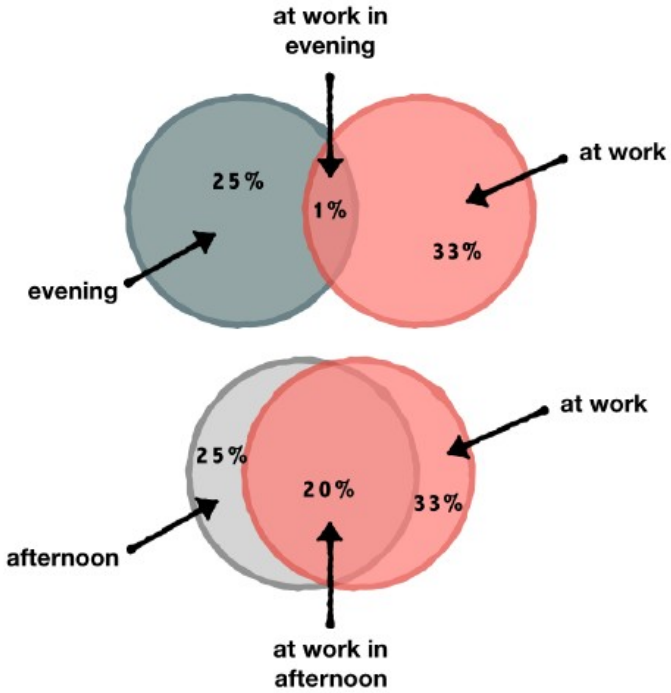
- ***Learning may be slow*** (depending on the parameterization)
- It is not possible to iteratively improve the generated models

E.g. Predict patients for high risks for certain diseases

NAIVE BAYES (SUPERVISED)

Classification technique based on Bayes' theorem (conditional probability and dependent events).





The **conditional probability** of events A and B is denoted P(A | B)

- P(A | B) = P(A and B) / P(B)
- P(work | evening) = 1 / 25 = 4%
- P(work | afternoon) = 20 / 25 = 80%

NAIVE BAYES (SUPERVISED)

Advantages:

- very easy to build and particularly useful for very large data sets.
- perform well for both binary and multi-class classifications.
- a good choice when CPU and memory resources are a limiting factor or if something fast and easy that performs pretty well is needed.

Disadvantages:

- Assume all the features are independent/unrelated, then cannot learn the interactions between features.

E.g.

- mining housekeeping genes
- genetic association studies
- discovering Alzheimer genetic biomarkers from whole genome sequencing (WGS) data

SUPPORT VECTOR MACHINES (SUPERVISED)

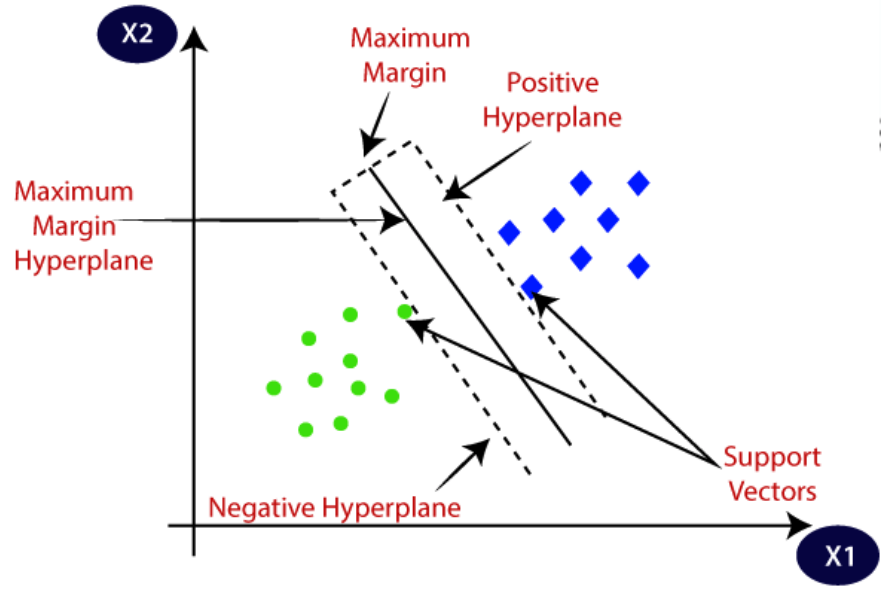
Used for both classification and regression problems, but primarily for classification

Classification: when the data has exactly two classes.

Goal: find the best decision boundary (hyperplan)

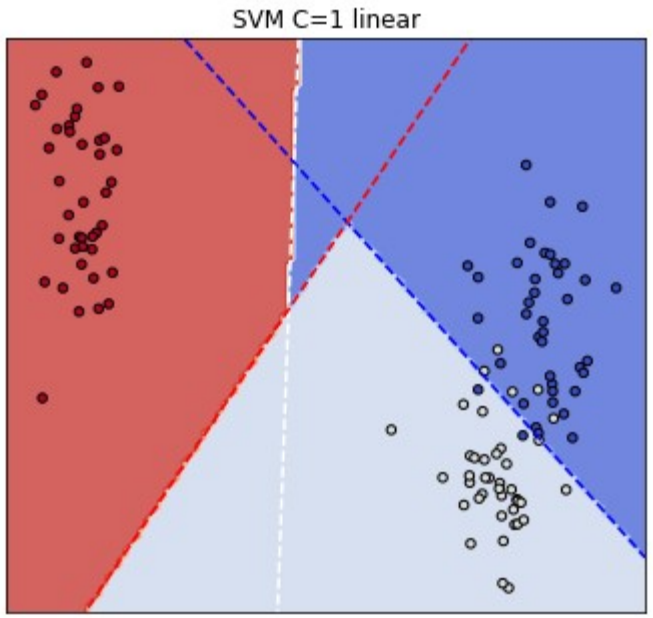
that differentiates the two classes

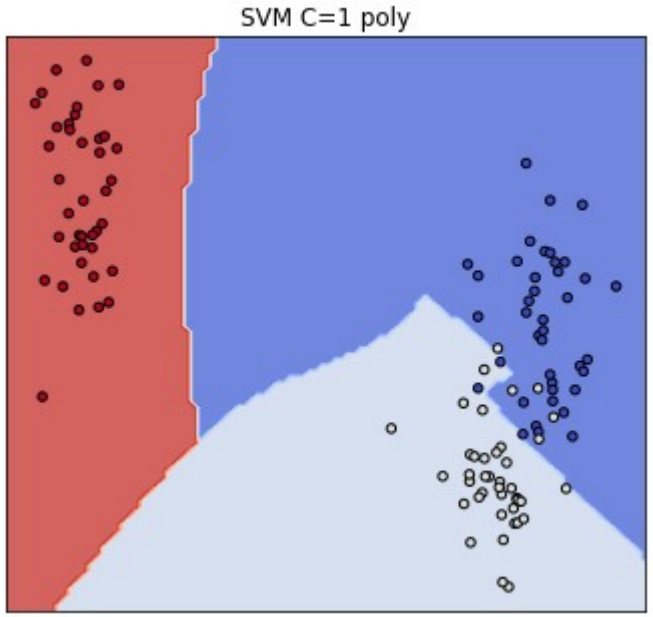
in n-dimensional space (n features)

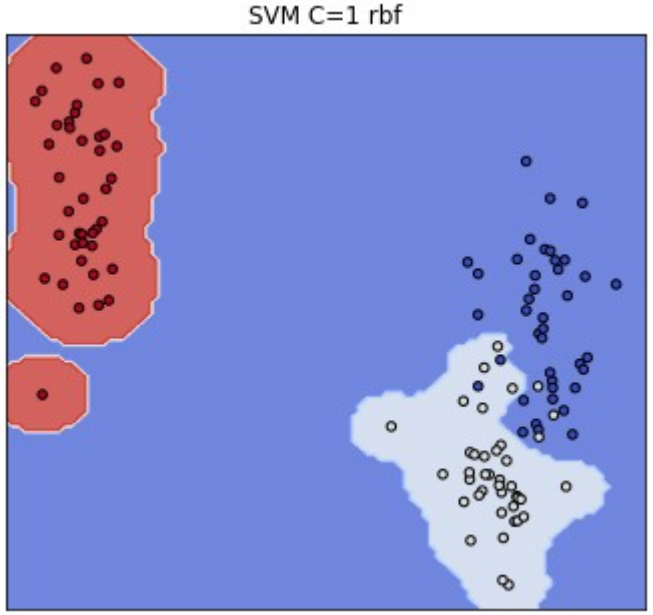


https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

SUPPORT VECTOR MACHINES (SUPERVISED)







SUPPORT VECTOR MACHINES (SUPERVISED)

Advantages:

- high accuracy
- high dimensional data
- work with both linearly and non-linearly separable data, with an appropriate kernel

Disadvantages:

- memory-intensive
- hard to interpret
- and difficult to tune.

E.g.

- Detecting common diseases such as diabetes
- Classification of genomic islands
- Classification of genes