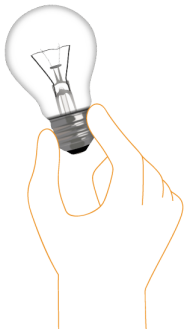
Machine Learning and Deep learning Methods

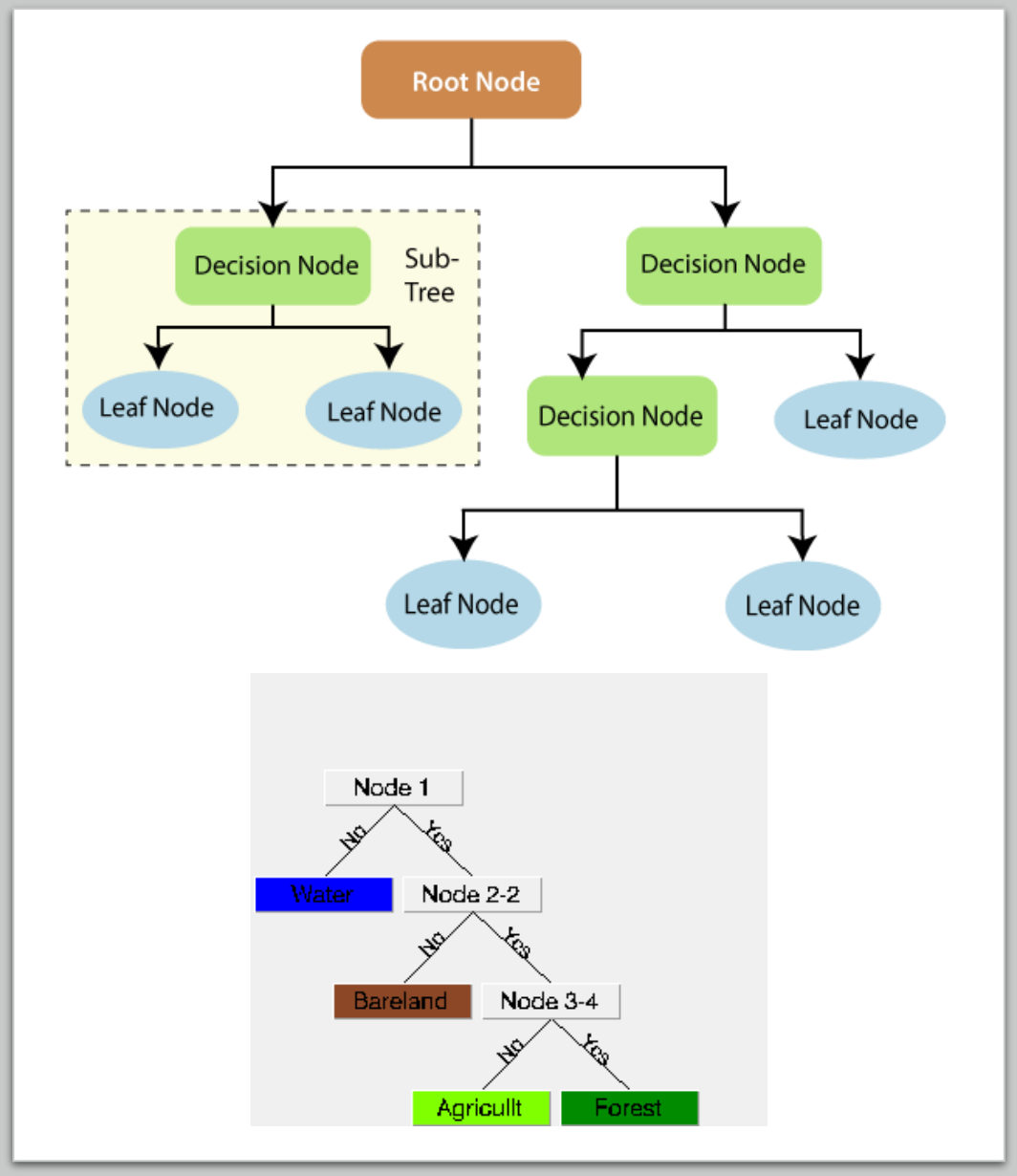


Bhogendra Mishra, PhD

Decision Tree

- One of the most powerful and popular tool for classification and prediction.
- It is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A great base model for machine learning, but not a great final model.



NDVI for land cover classification

Ensemble

- A single decision tree will rarely generalize well to data it wasn't trained on.
- We can combine the predictions of a large number of decision trees to make very accurate predictions.
- Mathematically speaking, a decision tree has low bias and high variance. Averaging the result of many decision trees reduces the variance while maintaining that low bias.
- Combining trees is known as an 'ensemble method'.
- Common ensemble methods include majority voting, weighted voting, and simple average.
- The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner.

Ensemble

- In **Majority voting** method, every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes. If none of the predictions get more than half of the votes, we may choose the most voted prediction but that considered as an unstable (weak) prediction.
- In **weighted voting** you count the prediction of the better models multiple times. Finding a reasonable set of weights is up to the user/expert.
- In **simple averaging** method, for every instance of test dataset, the average predictions are calculated. This method often reduces overfit and creates a smoother regression model.

Bootstrap

Bootstrapping is a resampling technique that involves repeatedly drawing samples from a source data with replacement, often to estimate a population parameter.
Replacement mean that the same data point may be included in our resampled dataset multiple times.

E.g., if our training set consists of 7 training samples, bootstrap samples (here: n=7) can look as follows, where C1, C2, … Cm shall symbolize the decision tree classifiers.

Bagging

Bootstrapping: Bagging uses a bootstrapping sampling technique to create diverse samples. This resampling method generates different subsets of the training dataset by selecting data points at random and with replacement.

Parallel training: These bootstrap samples are then trained independently and in parallel with each other using weak or base learners.

Aggregation: Finally, depending on the task (i.e. regression or classification), an average or a majority of the predictions are taken to compute a more accurate estimate. In the case of regression, an average is taken of all the outputs predicted by the individual classifiers; this is known as soft voting. For classification problems, the class with the highest majority of votes is accepted; this is known. as hard voting or majority voting

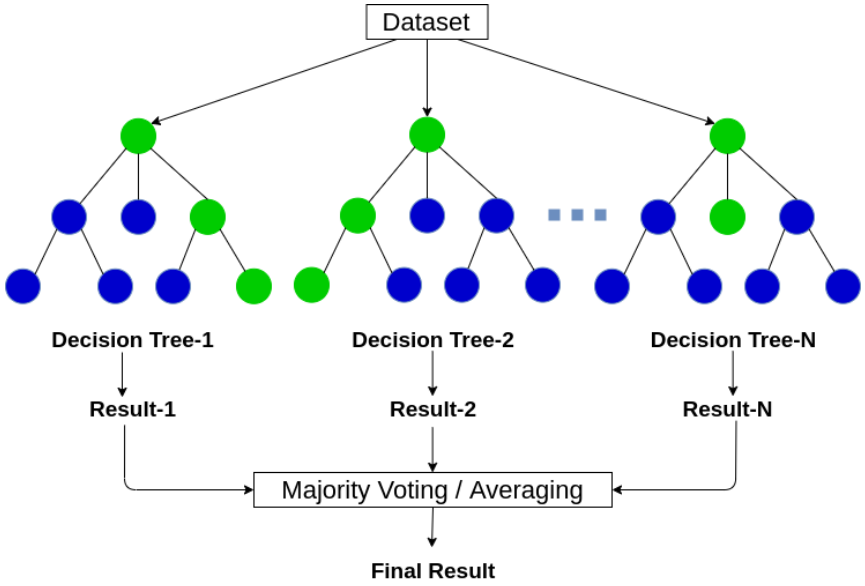
Random Forest

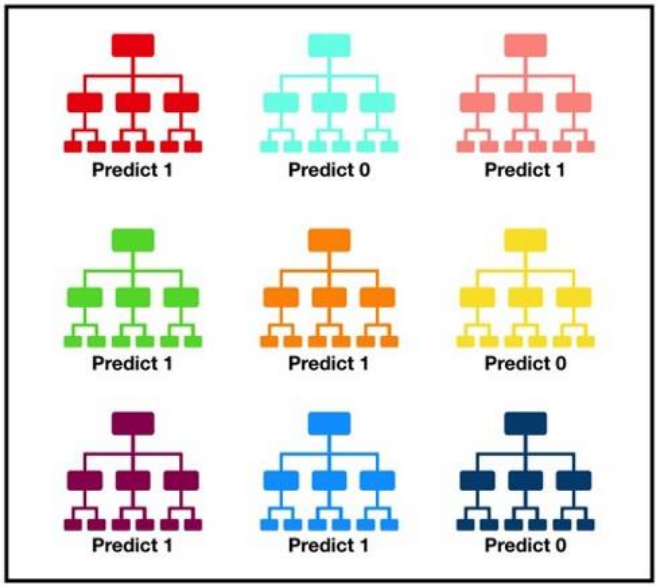
- A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.
- A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating.
- The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.
- A random forest eradicates the limitations of a decision tree algorithm.

Random Forest

In random forest, we build a number of decision trees on bootstrapped training samples each time a split in a tree is considered, a random sample of *m* predictors is chosen as split candidates from the full set of *p* predictors. Random forest improves on bagging because it decorrelates the trees with the introduction of splitting on a random subset of features.

Note that if *m* = *p*, then this is bagging.

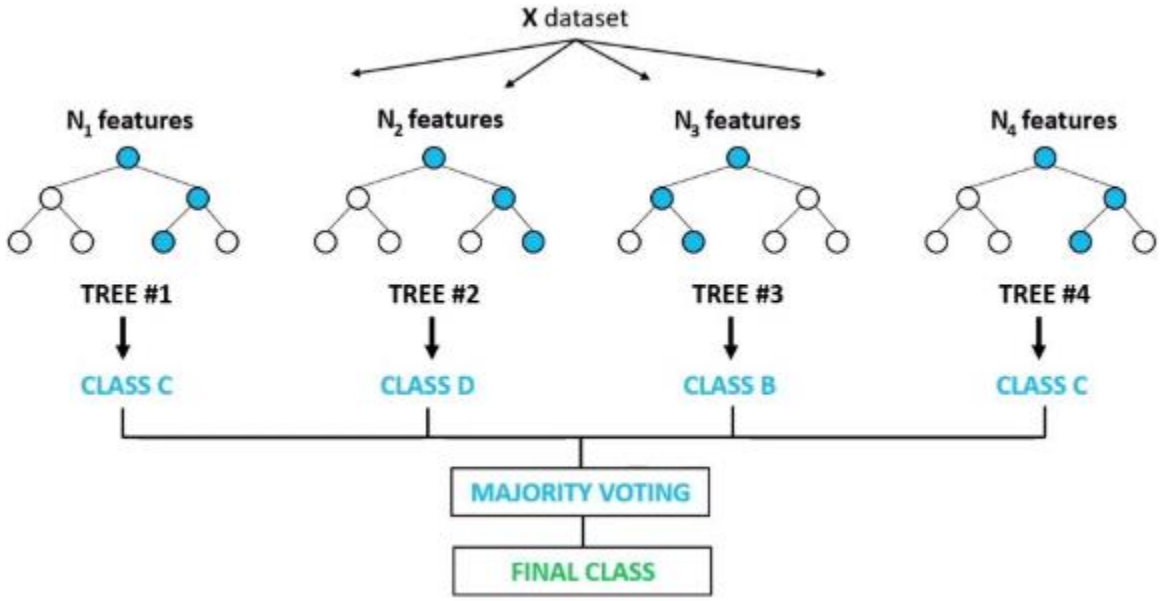




Tally: Six 1s and Three 0s

Prediction: 1

Random Forest Classifier



- **Step 1**: The algorithm selects random samples from the dataset provided.
- **Step 2:** The algorithm will create a decision tree for each sample selected. Then it will get a prediction result from each decision tree created.
- **Step 3: V**oting will then be performed for every predicted result. For a classification problem, it will use **mode**, and for a regression problem, it will use **mean**.
- **Step 4**: And finally, the algorithm will select the most voted prediction result as the final prediction.

Random Forest

For b = 1 to B:

(a) Draw a bootstrap sample Z^* of size *N* from the training data.

(b) Grow a random-forest tree to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $\boldsymbol{n_{min}}$ is reached.

i. Select ***m*** variables at random from the *p* variables.

ii. Pick the best variable/split-point among the *m*.

iii. Split the node into two daughter nodes.

Output the ensemble of trees.

To make a prediction at a new point *x* we do:

For regression: average the results

For classification: majority vote

Random Forests Tuning

The inventors make the following recommendations:

- For classification, the default value for *m* is $\sqrt{p}$ and the minimum node size is one.
- For regression, the default value for m is $p/3$ and the minimum node size is five.

In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters.

Like with Bagging, we can use OOB and therefore RF can be fit in one sequence, with cross-validation being performed along the way. Once the OOB error stabilizes, the training can be terminated.

Performance estimation

- For each bootstrap sample taken from the training data, there will be samples left behind that were not included. These samples are called **Out-Of-Bag samples or OOB**.
- The performance of each model on its left-out samples when averaged can provide an estimated accuracy of the bagged models. This estimated performance is often called the **OOB estimate of performance**.
- These performance measures are reliable test error estimate and correlate well with cross validation estimates.

Variables important

- As the Bagged decision trees are constructed, we can calculate how much the error function drops for a variable at each split point.
- In regression problems this may be the drop in sum squared error and in classification this might be the Gini score.
- These drops in error can be averaged across all decision trees and output to provide an estimate of the importance of each input variable. The greater the drop when the variable was chosen, the greater the importance.

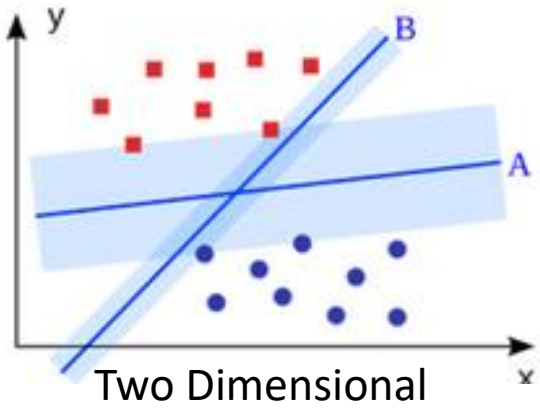
The mean decrease in **Gini coefficient** is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest.

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

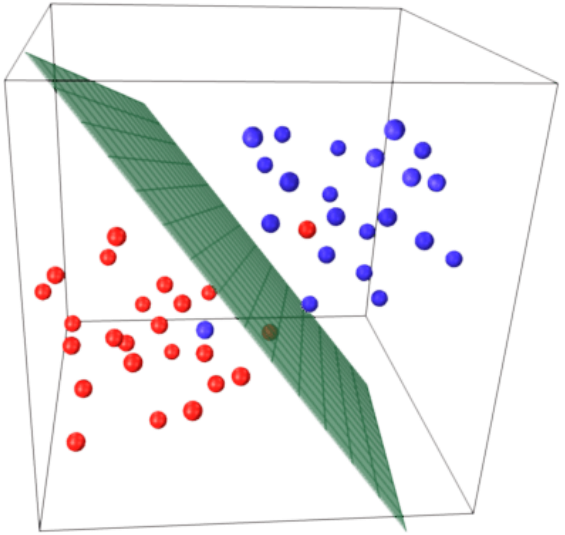
Where Pi denotes the probability of an element being classified for a distinct class.

Support Vector Machine

- A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. Through the input training dataset, the algorithm identify an optimal hyperplane which categorizes dataset into class.
- In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

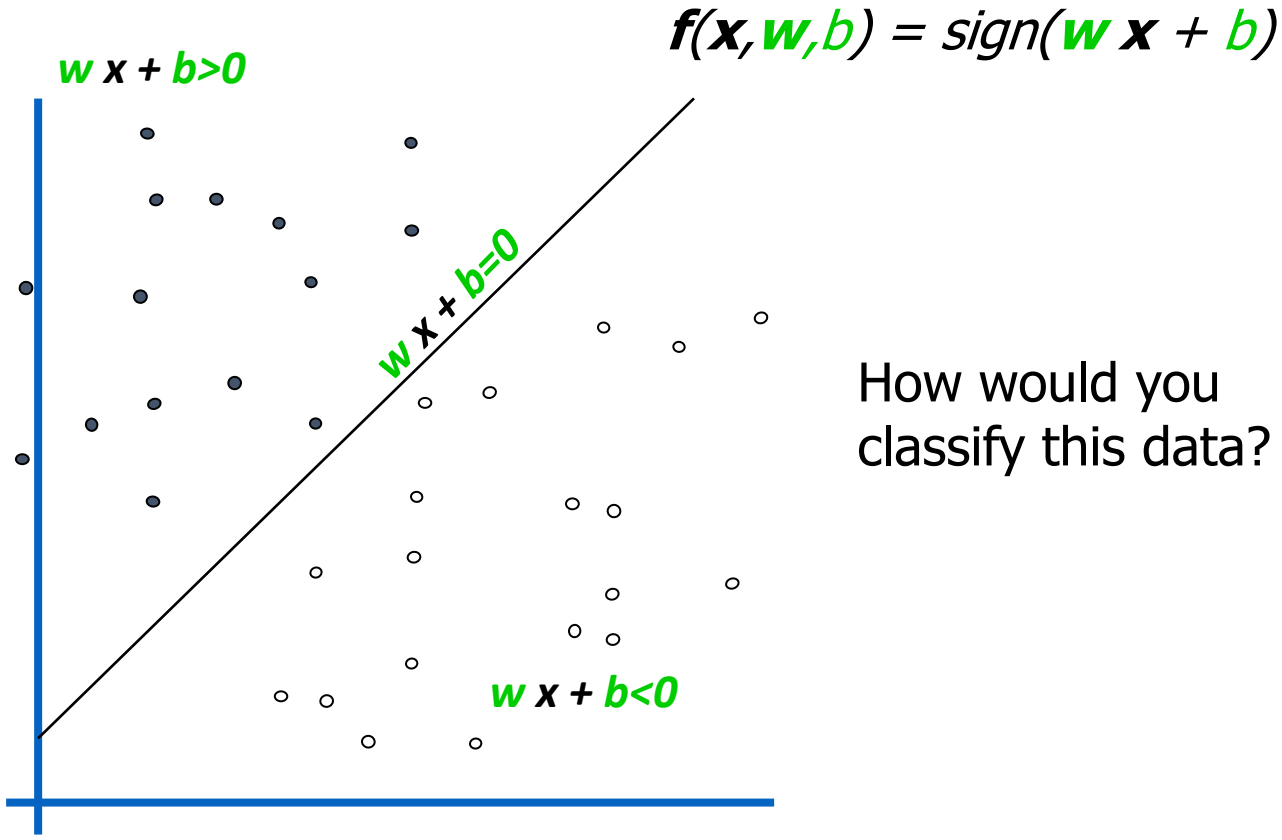


Two Dimensional


Three Dimensional

Linear Classifiers

- denotes +1
- denotes -1

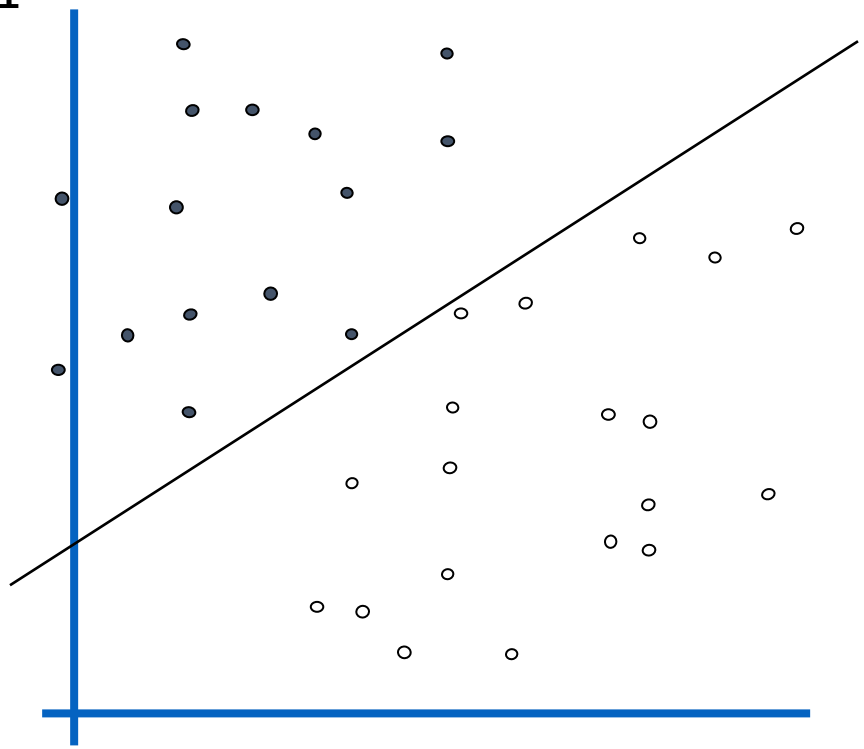


How would you classify this data?

Linear Classifiers

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

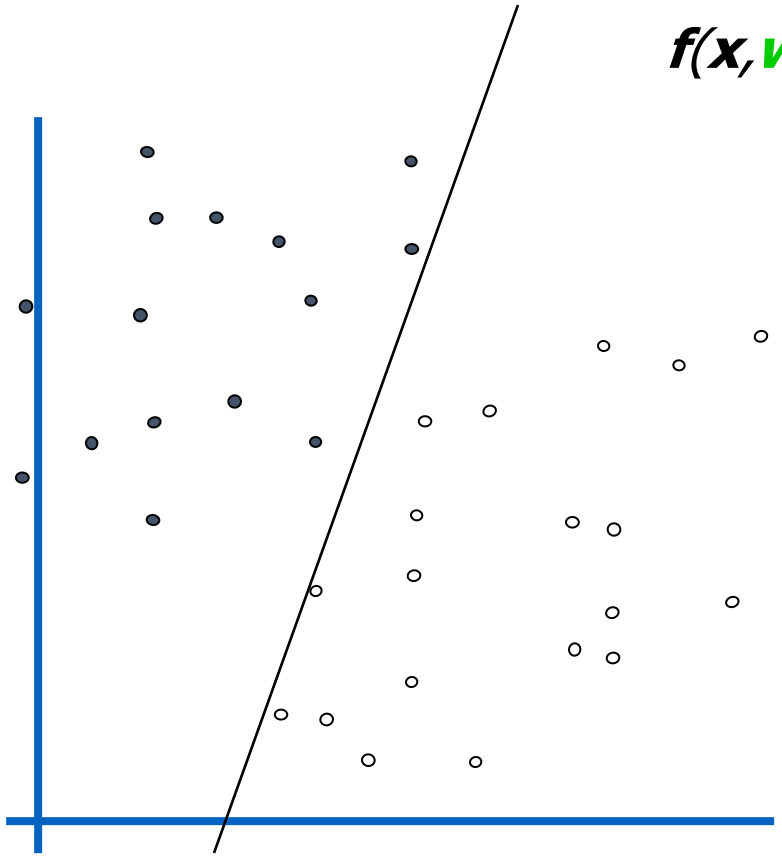
- denotes +1
- denotes -1



How would you classify this data?

Linear Classifiers

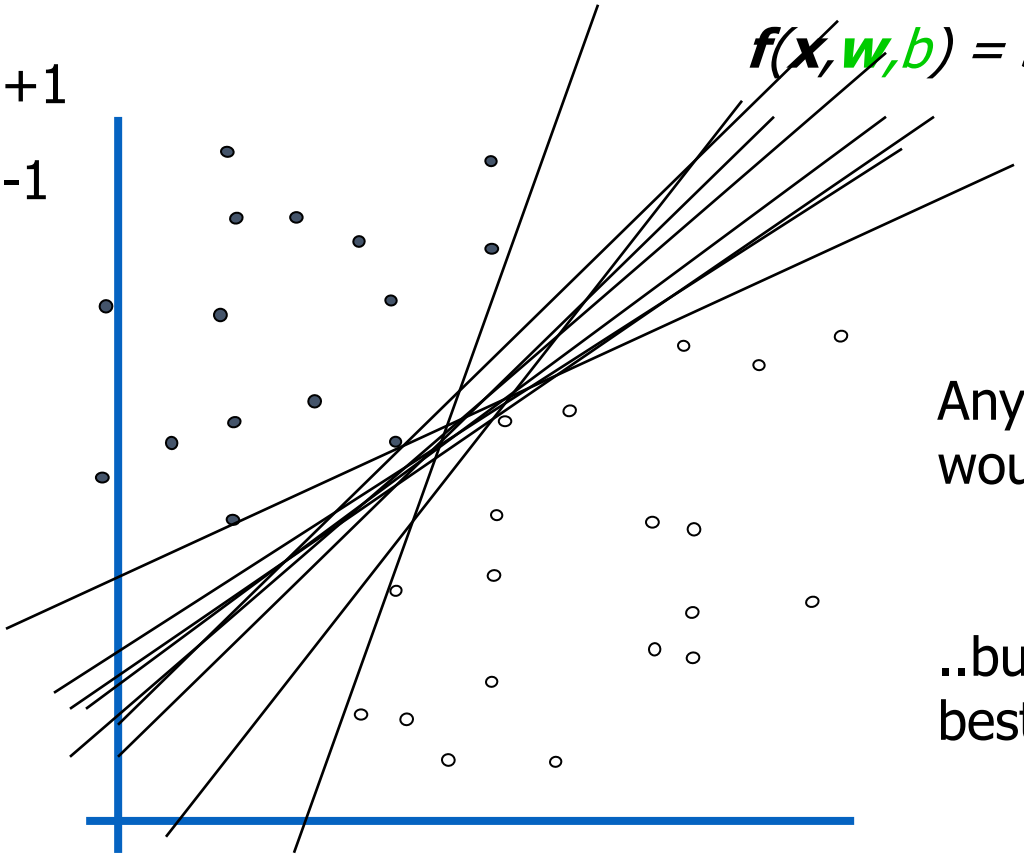
- denotes +1
- denotes -1



$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$$

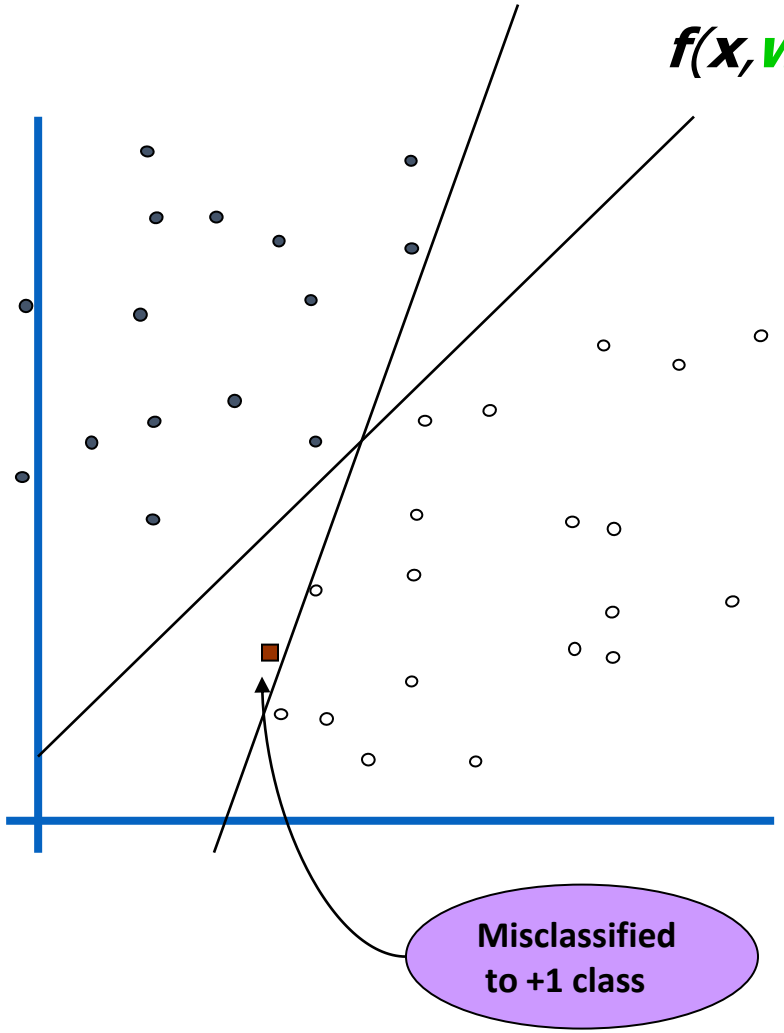
How would you classify this data?

Linear Classifiers



Linear Classifiers

- denotes +1
- denotes -1

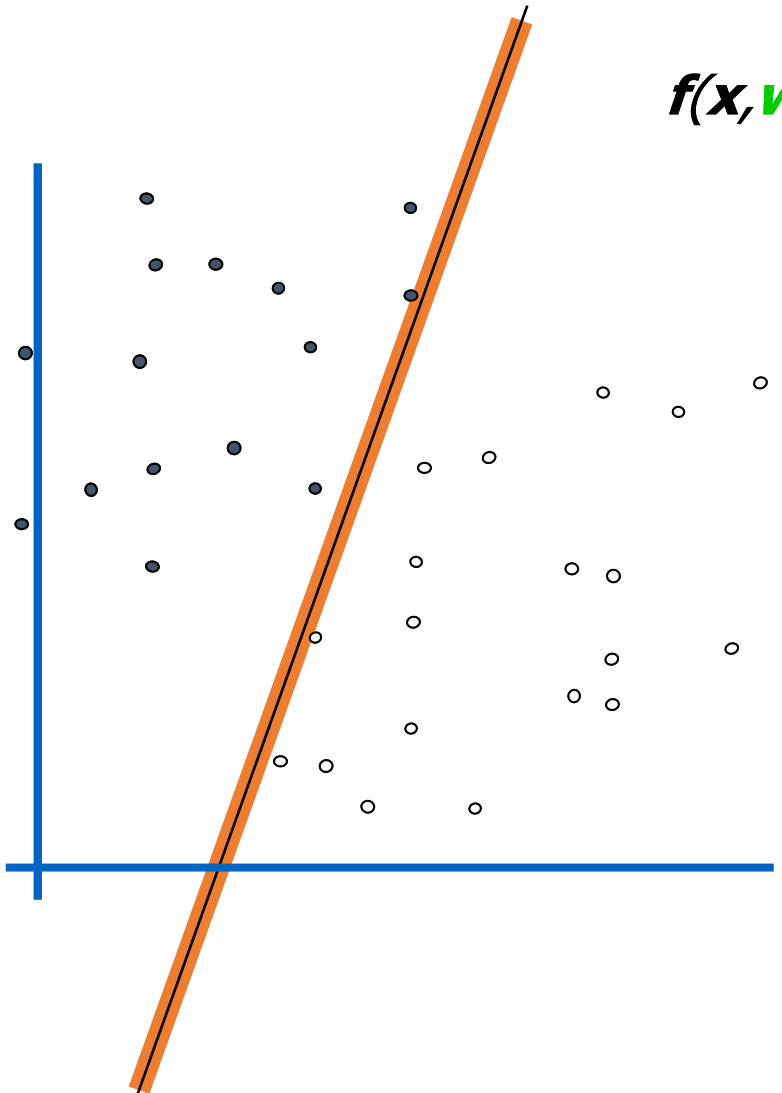


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

How would you classify this data?

Classifier Margin

- denotes +1
- denotes -1



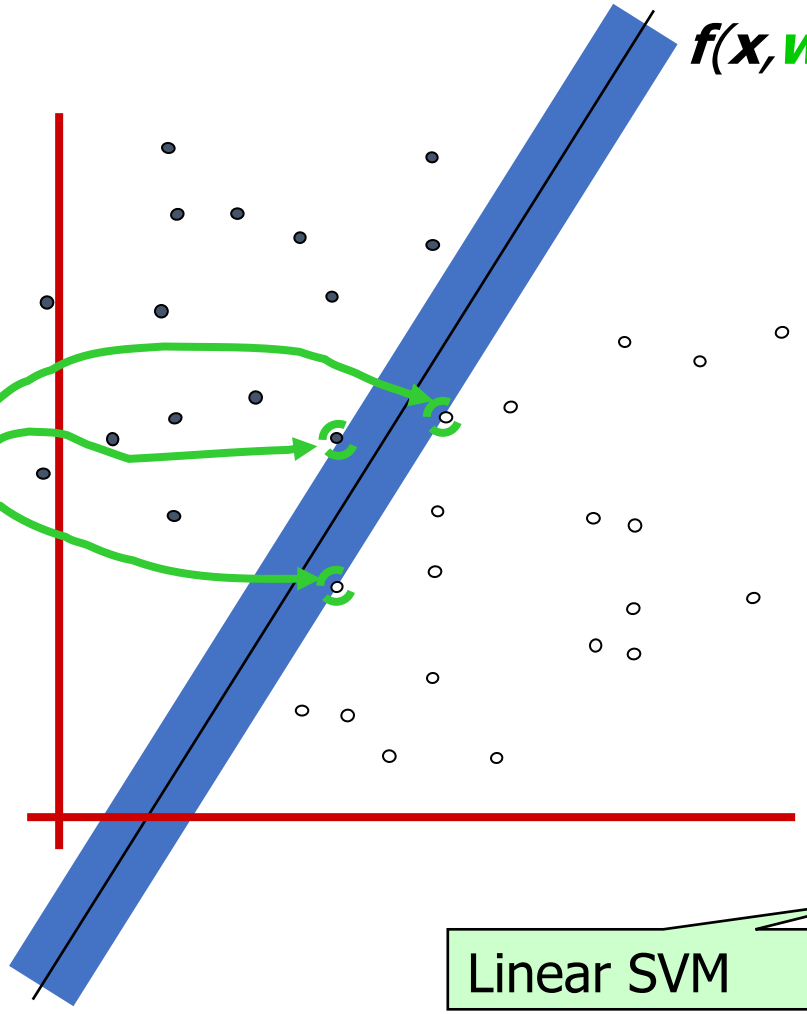
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin

- denotes +1
- denotes -1

$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$

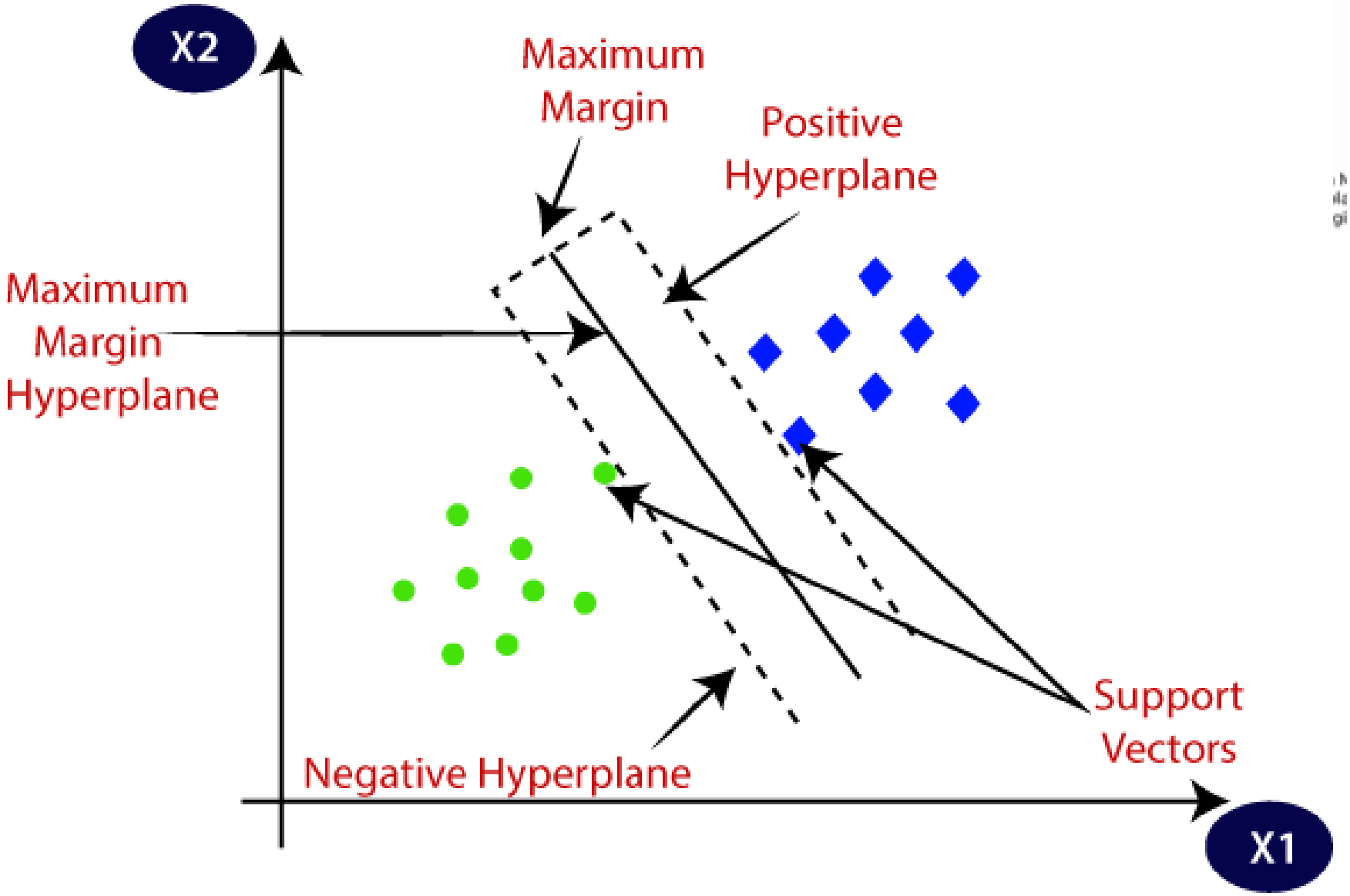


Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear SVM


X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

Nonlinear SVM - Overview

- SVM locates a separating hyper-plane in the feature space and classify points in that space
- It does not need to represent the space explicitly, simply by defining a kernel function
- We can have different types of kernel functions.
- The linear kernel function plays the role of the dot product in the feature space.

Feature Space
Feature space refers to the n-dimensions where your variables live (not including a target variable, if it is present).

Kernel Function Example

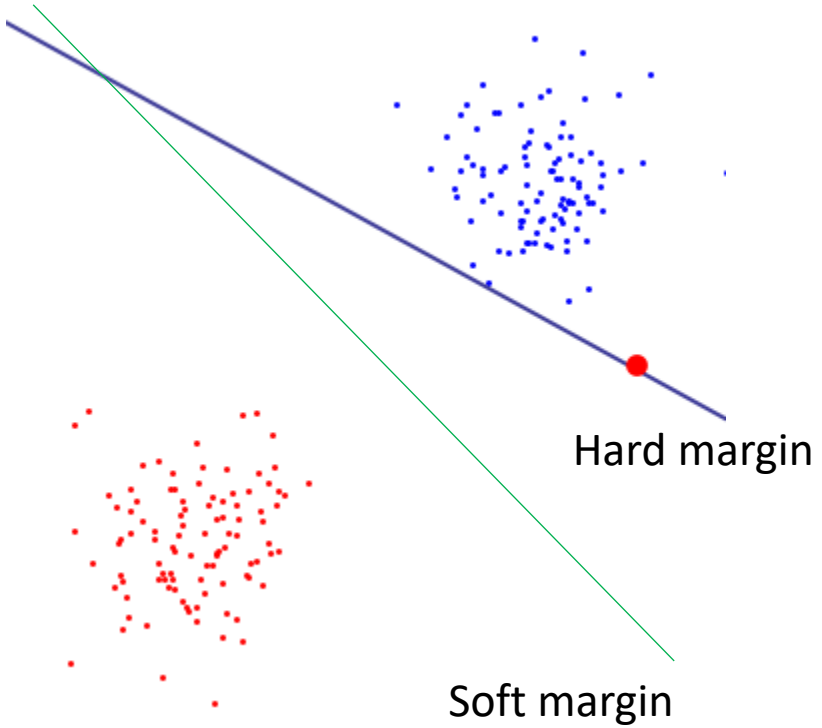
- Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^T\mathbf{x_j}$
- Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j}) = (1+ \mathbf{x_i}^T\mathbf{x_j})^p$
- Gaussian (radial-basis function network):

$$K(\mathbf{x_i},\mathbf{x_j}) = \exp\left(-\frac{\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0\mathbf{x_i}^T\mathbf{x_j} + \beta_1)$

Soft and Hard Margin

- If training data is linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible.
- The region bounded by these two hyperplanes is called the "margin".
- If it allows to have an exception – small error in separating the classes, is soft margin, while it does not tolerate even by a single dataset, is called hard margin



How SVM works
https://www.youtube.com/watch?v=1NxnPkZM9bc

Properties of SVM

- Flexibility in choosing a similarity function
- Sparseness of solution when dealing with large data sets
  - only support vectors are used to specify the separating hyperplane
- Ability to handle large feature spaces
  - complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution
- Feature Selection