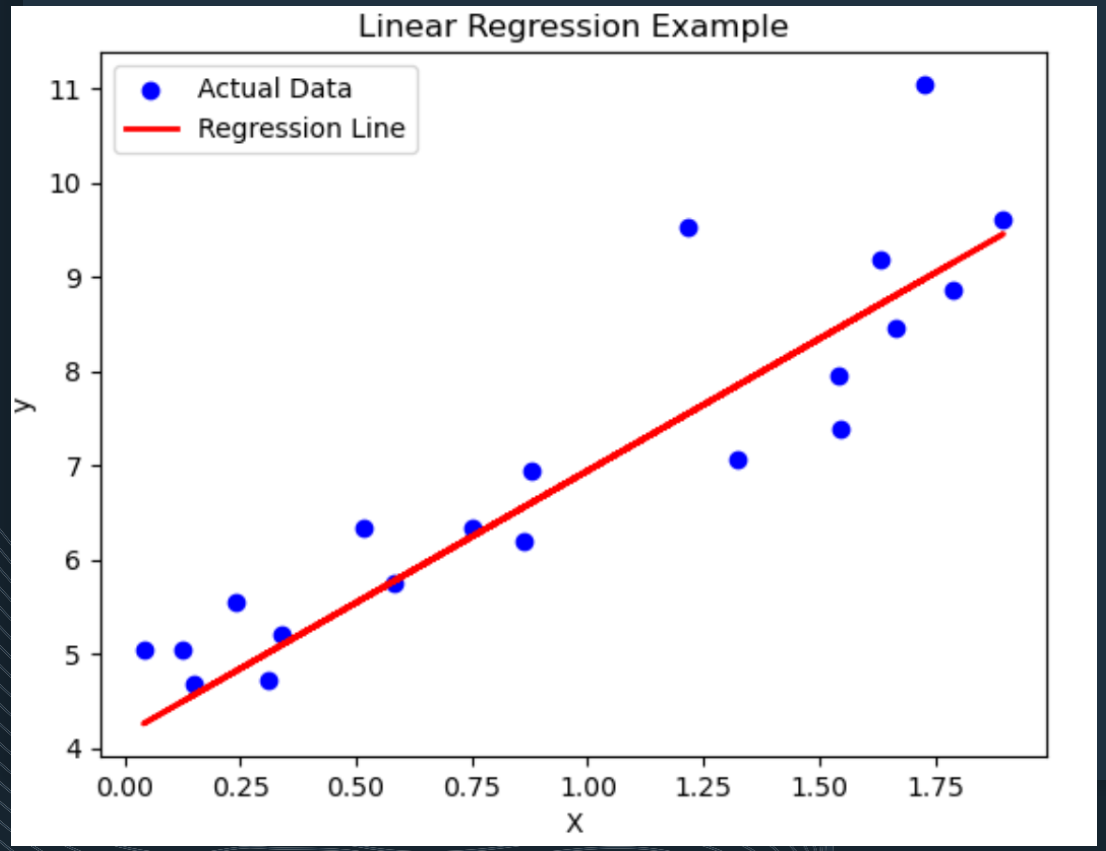
Classification

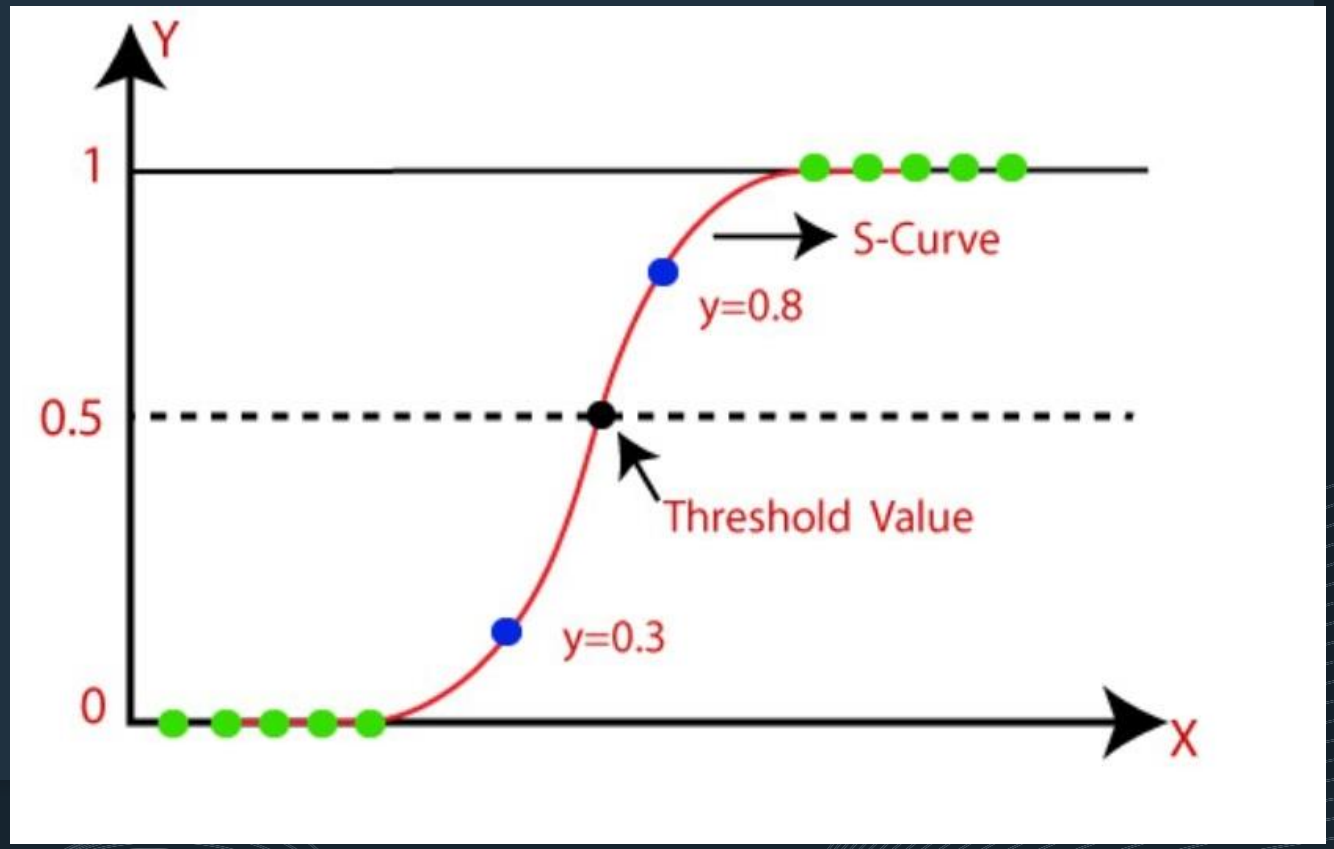
Yao Zhang - Aalto University

Expected Outcomes

- Try different classification methods
- linear regression, logistic regression, k-fold cross validation, KNN, SVM, different performance matrics

Linear regression and logistic regression

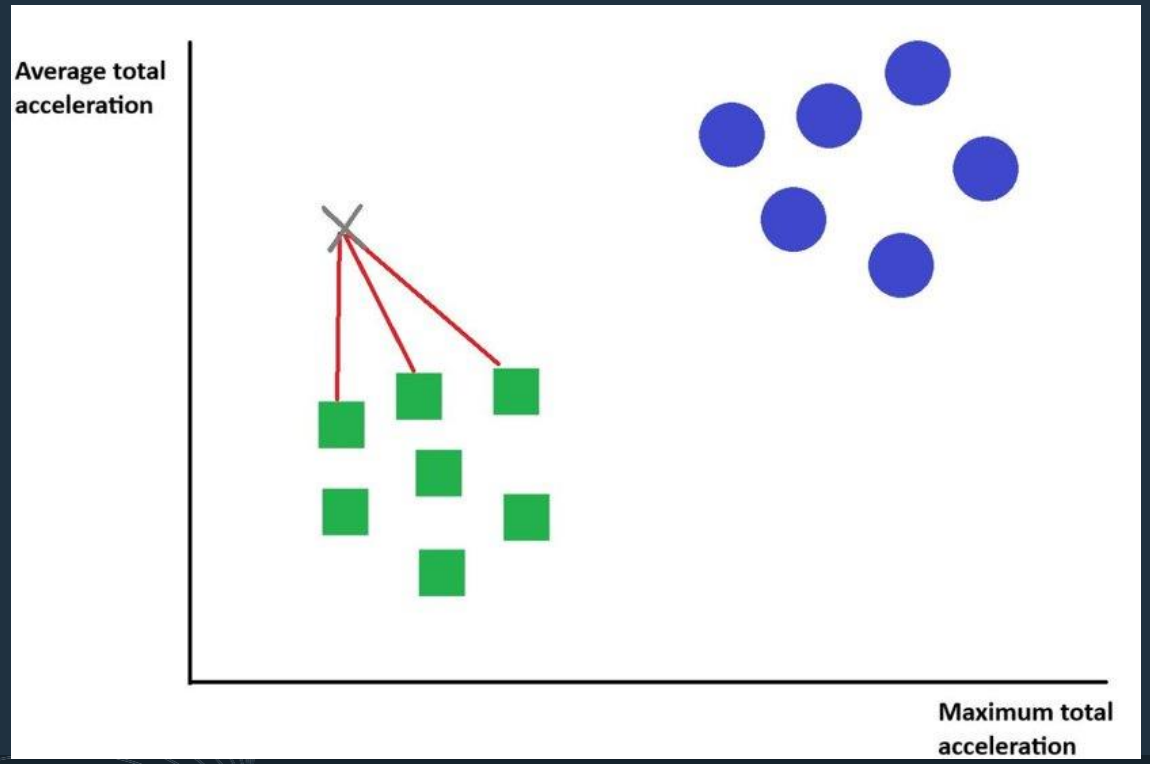




Linear regression is a method for finding the straight line or hyperplane that best fits a set of points.

Logistic regression predicts the probability of an event taking place based on the input.

k-Nearest Neighbors(KNN)



KNN works by finding the K-nearest neighbors to the sample we want to predict the class

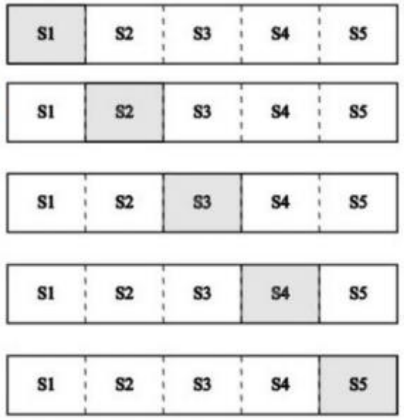
Frequently fine-tune hyper-Parameter: n_neighbors, weights, 'p': [1, 2] # method for measure distance(p=1 for Manhattan distance, p=2 for Euclidean distance)

k-fold cross validation

Split the Data: Divide your dataset into K equal-sized subsets (folds).

Iterate: Use each subset as a validation set and the remaining K-1 subsets as the training set, then iterate this process K times.

Evaluate: Calculate the average performance (e.g., accuracy, precision, recall) across all folds for each candidate k (number of nearest neighbour).



Train on S2, S3, S4, S5, test on S1 → ε_1

Train on S1, S3, S4, S5, test on S2 → ε_2

⋮ ⋮

Train on S1, S2, S3, S4, test on S5 → ε_5

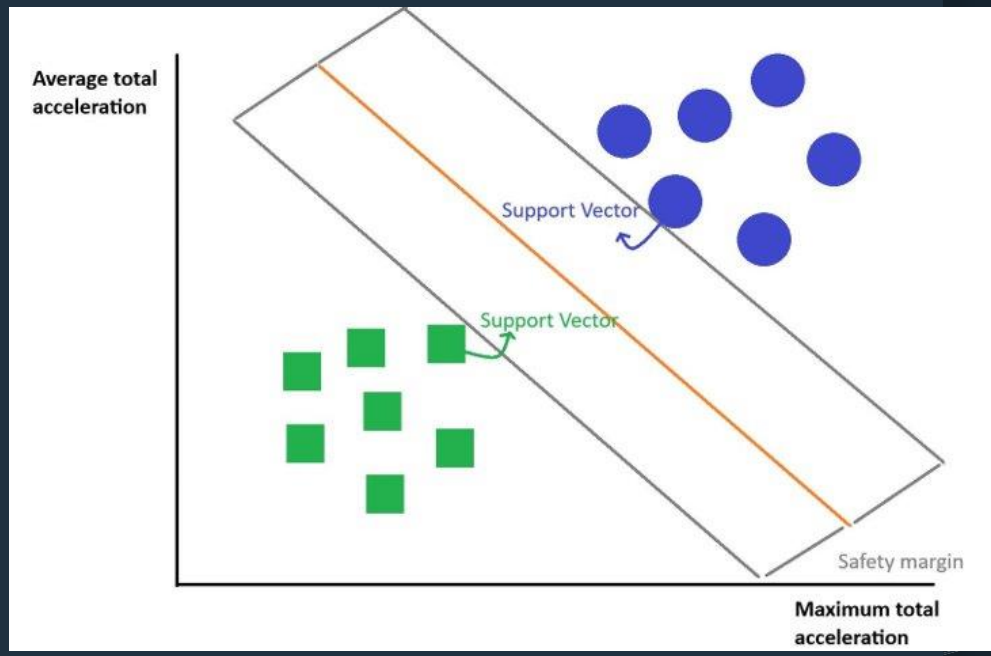
$$\varepsilon = \frac{1}{5}\sum_{i=1}^{5}\varepsilon_i$$

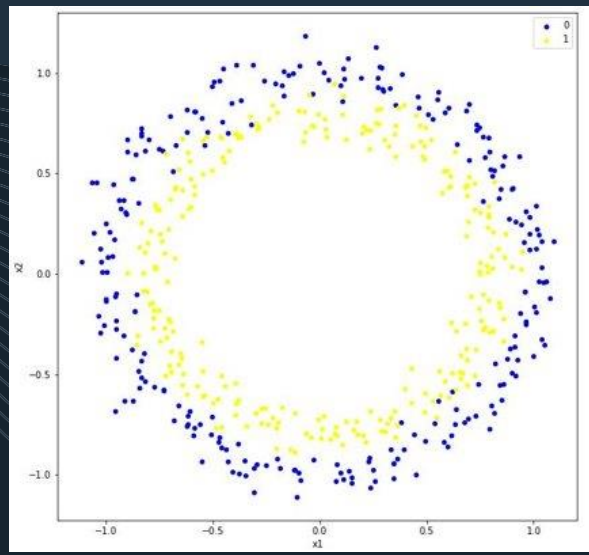
For selecting hyperparameter for training model

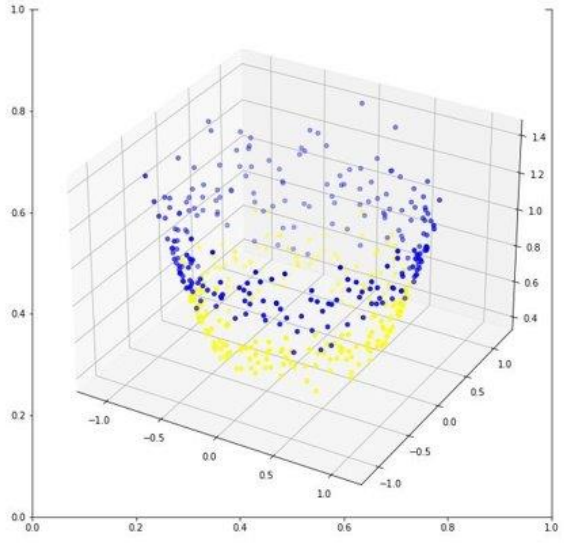
Support Vector Machine

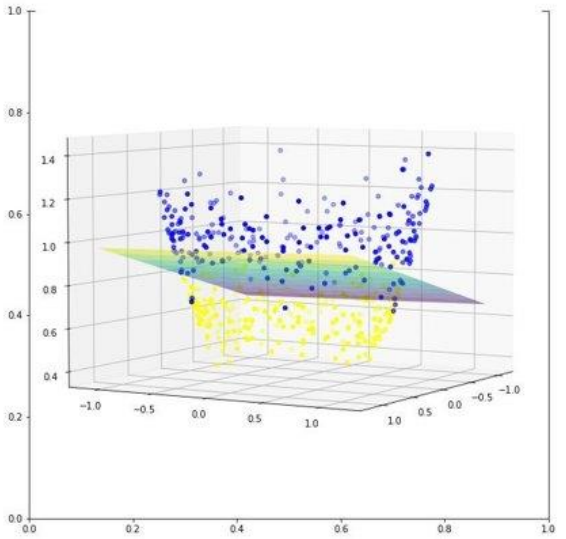
Support Vector Machines draw a **line** to divide these two classes.

How to separate linearly inseparable data?
Kernel trick: map the feature vectors into a higher dimensional space (**polynomial** and **radial basis)**





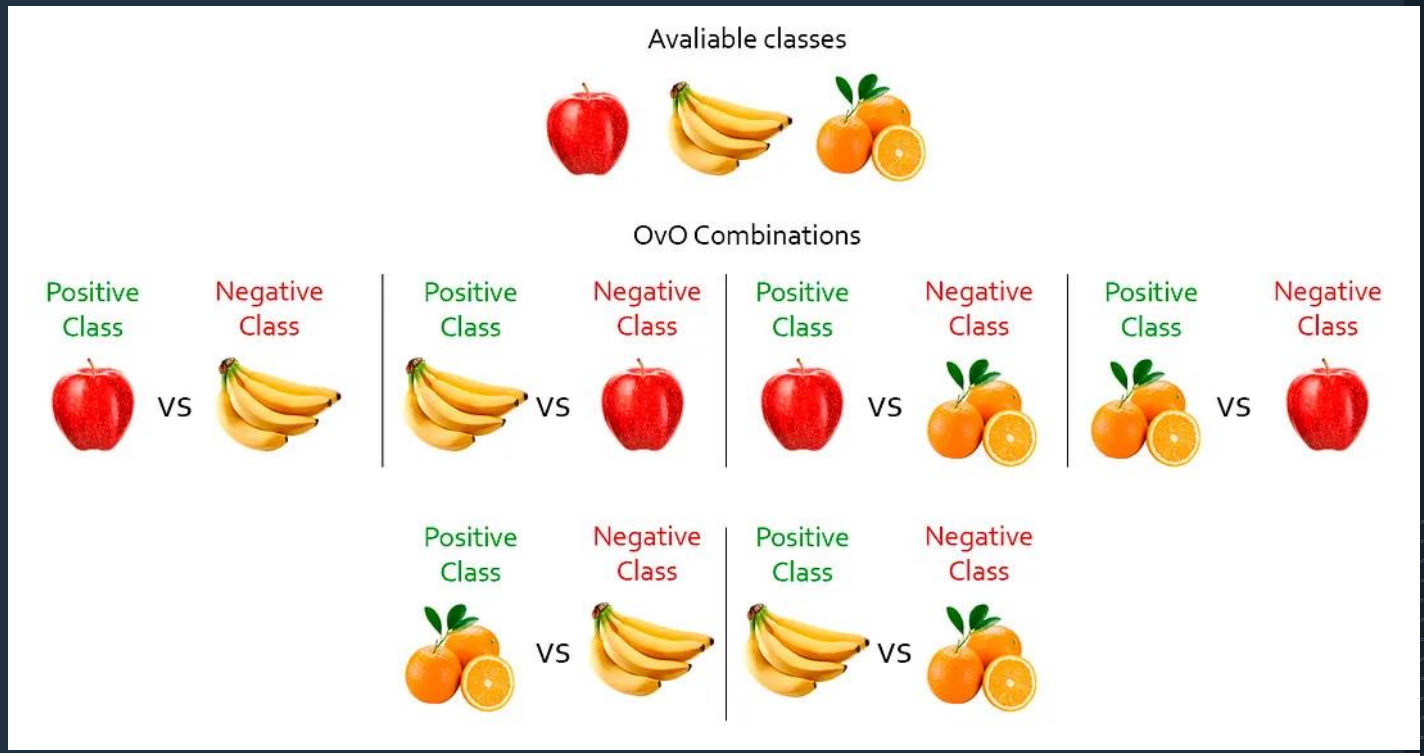




Support Vector Machine for multi-class classification

We have two strategies for that:

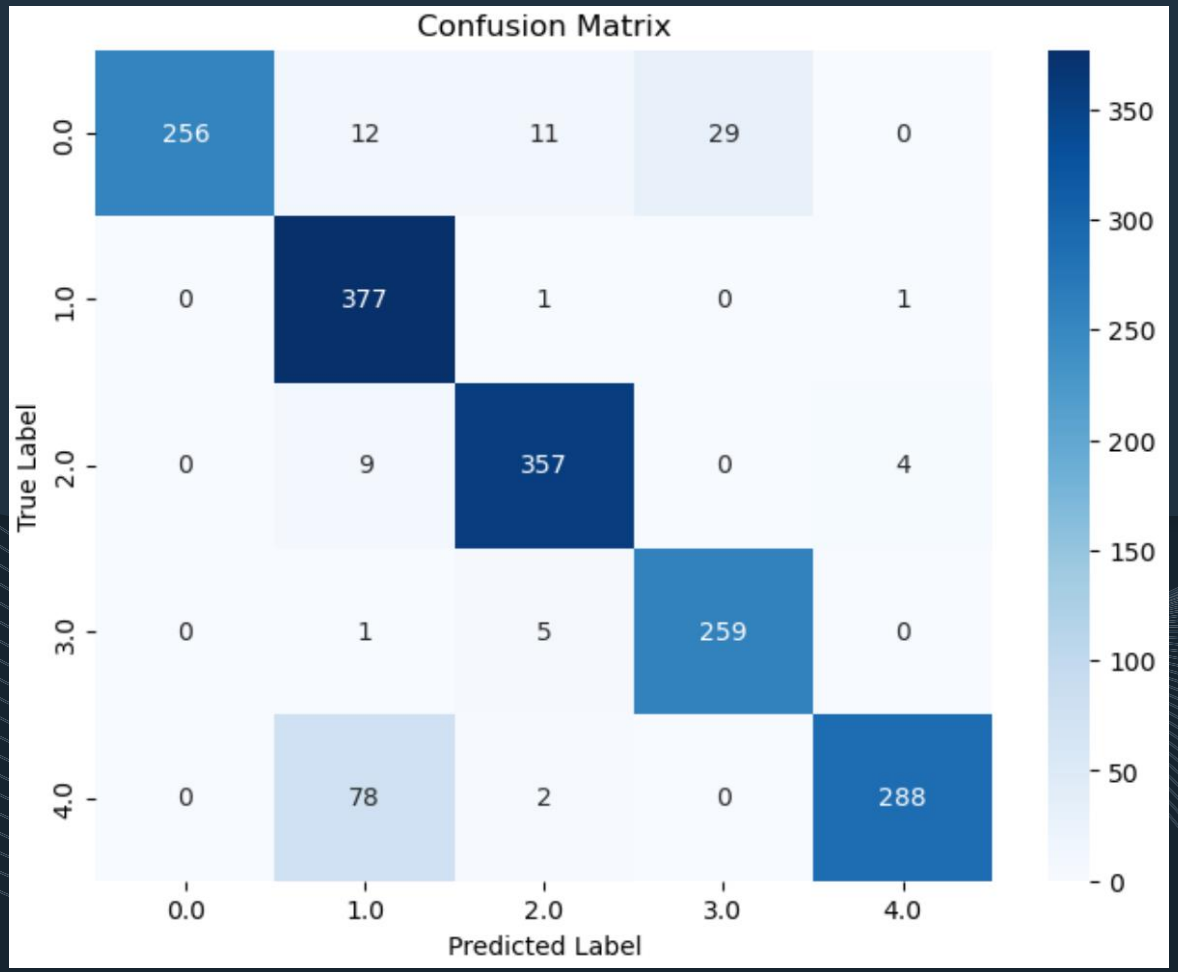
One-vs-One(OvO),One-vs-All (OvA)



Source:
https://www.kaggle.com/code/taweilo/ml-multiclass-ovo-ovr-note

Performance metrics

Confusion matrix, accuracy, precision, recall, F1 score



Accuracy: The ratio of correctly predicted instances to the total instances.

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

Precision: The ratio of correctly predicted positive observations to the total predicted positives.

$$Precision = \frac{TP}{TP+FP}$$

Recall: The ratio of correctly predicted positive observations to all the observations in the actual class.

$$Recall = \frac{TP}{TP+FN}$$

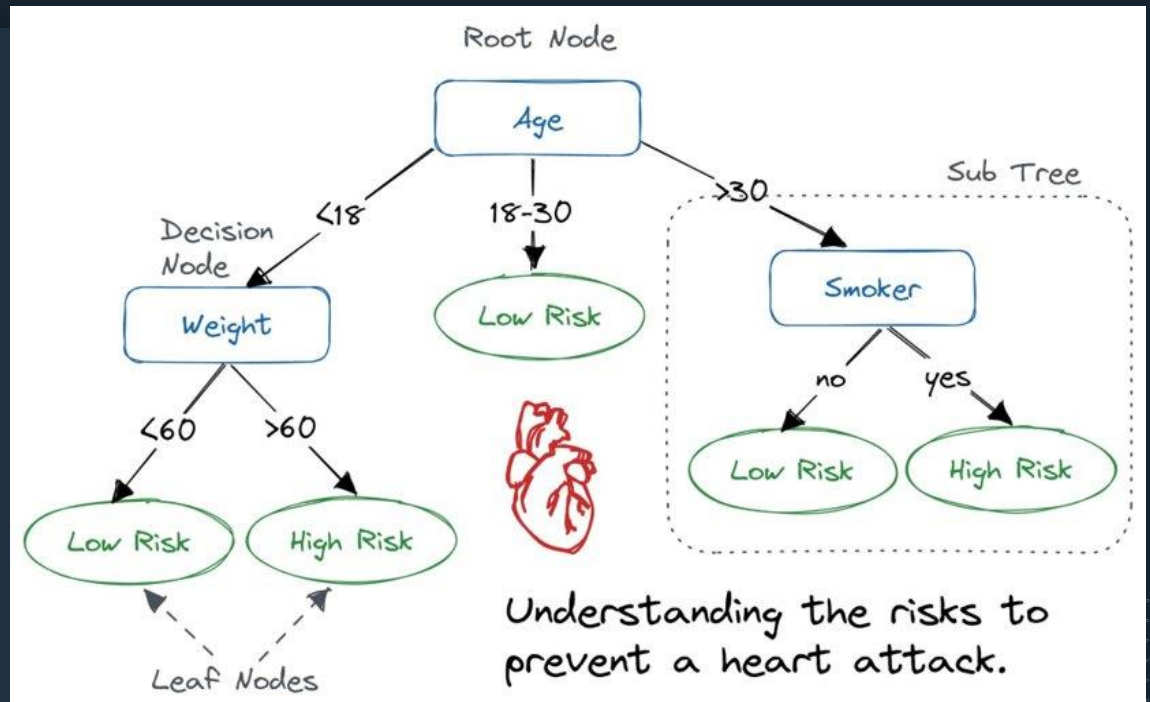
F1 score: The harmonic mean of Precision and Recall. Balances precision and recall, useful for uneven class distributions.

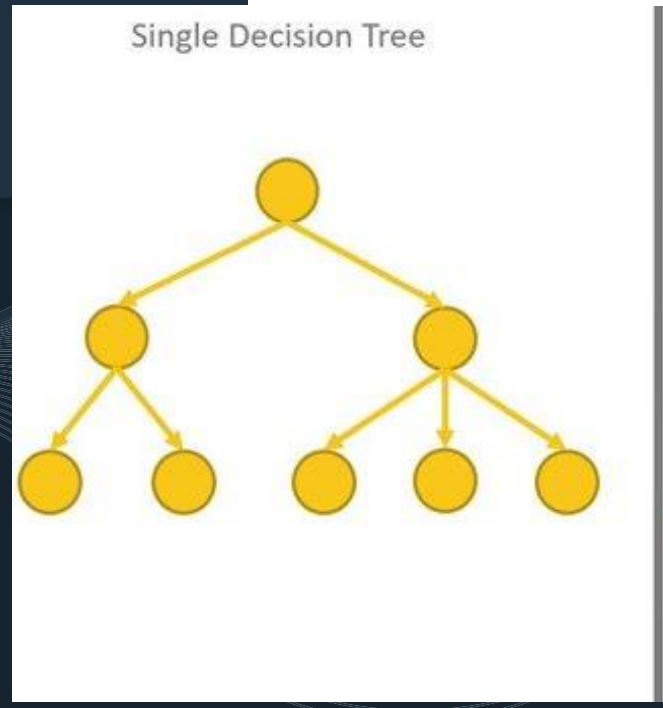
$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

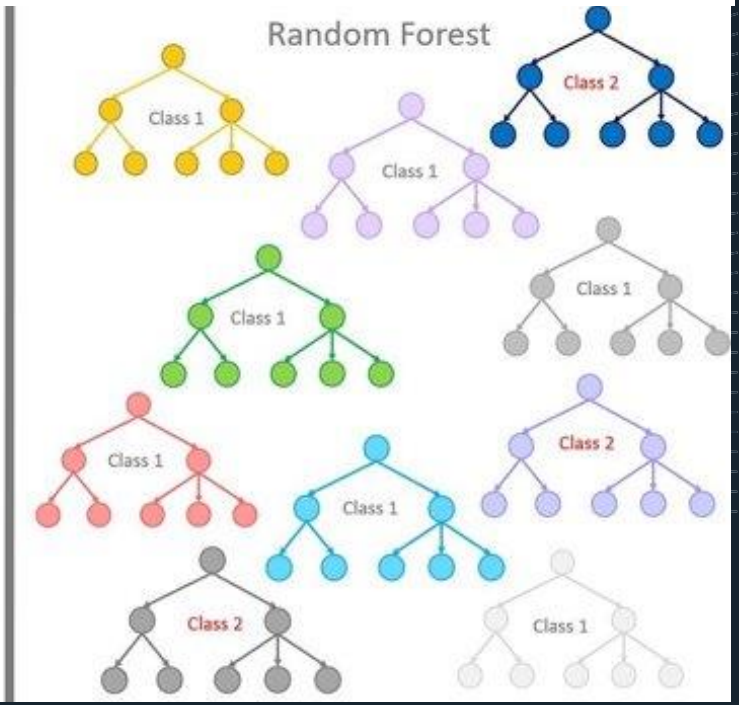
Decision Trees and Random Forest

The process of decision tree starts at the root node, classification results are based on the leaf node.

Random Forest is a group of decision trees







Decision Trees and Random Forest

We select different subsets from original data(**bootstrapping**)

Training a decision tree based on each subset

Combining predictions of different decisi on trees is called **aggregating**

Evaluating the performance of Random Forests based on those data(out-of-bag) which are never been selected by bootstrapping, **OOB-score**