



Artificial Intelligence and Machine Learning (ACCA6011)

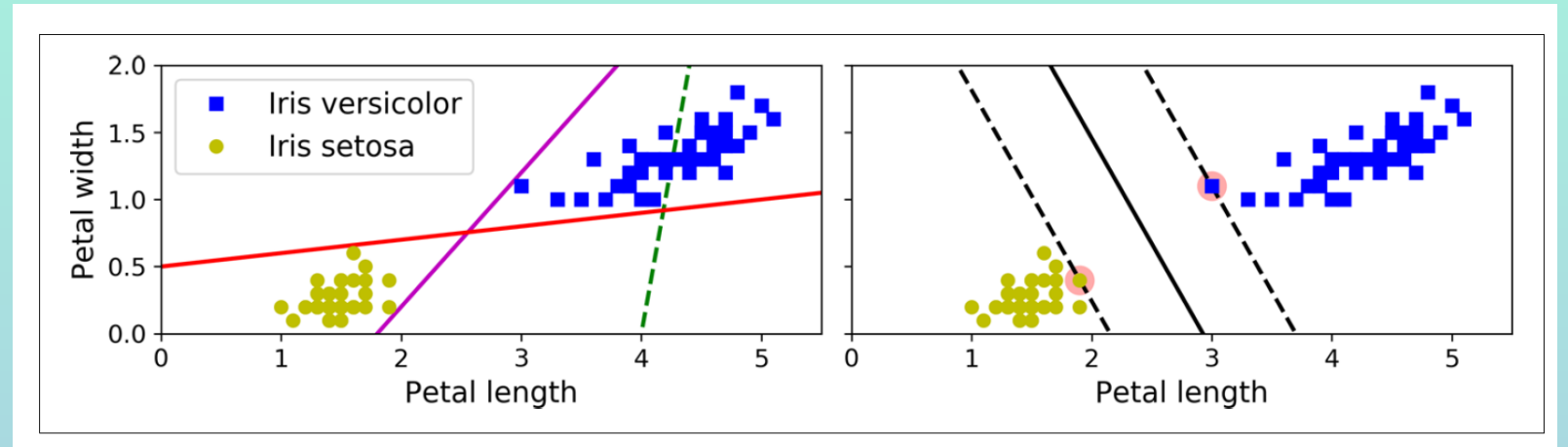
Supervised learning – Part 2

Dr. Seena Joseph

Supervised Learning

Support Vector Machine

Powerful ML algorithm for linear and non-linear classification and regression.



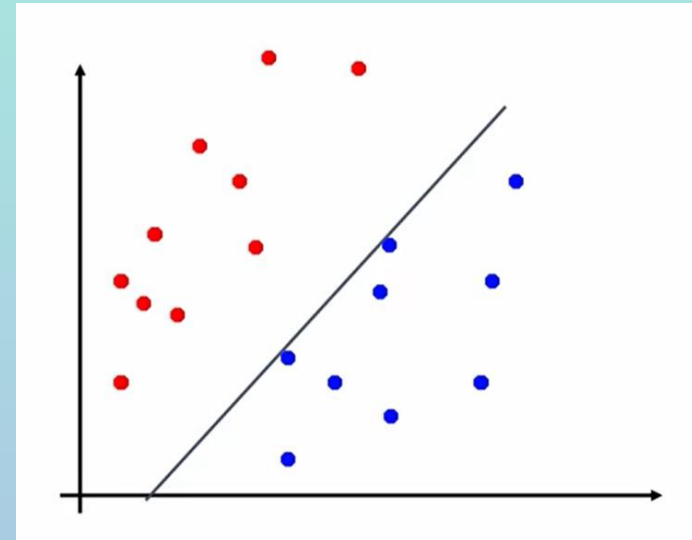
Example :

- The classes given here can be linearly separable.
- Green dashed line on the left image is not a good classifier as it does not separate the classes properly.
- Purple and Red lines are work for the training set, but the decision boundaries are close to the instance.
- The black line on the right image is a good model with a wide margin.

Supervised Learning

Support Vector Machine Refers to set of classifiers:

1. Maximal Margin Classifier [Linear Classifier]
2. Support Vector Classifier [Linear Classifier]
3. Support Vector Machine [Non-Linear Classifier]



Supervised Learning

Maximal Margin Classifier

Attempts to separate data point in d dimension into separate classes using hyperplanes.

In d dimension, hyperplane is a flat affine subspace of $d-1$ dimension.

- In 2-dimensional space the hyperplane is a line (Figure a)
- In 3- dimensions space the hyperplane is a plane (Figure b)
- Harder to visualise in higher dimensions

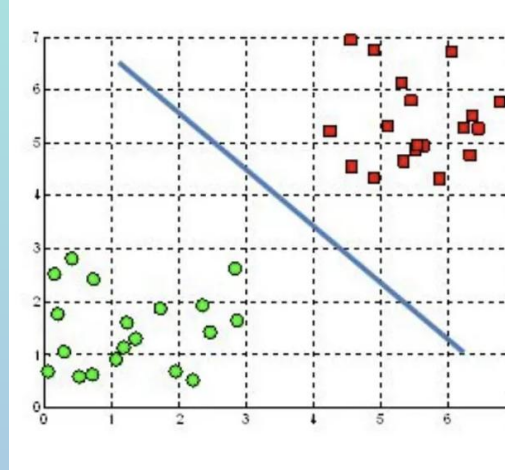


Figure a

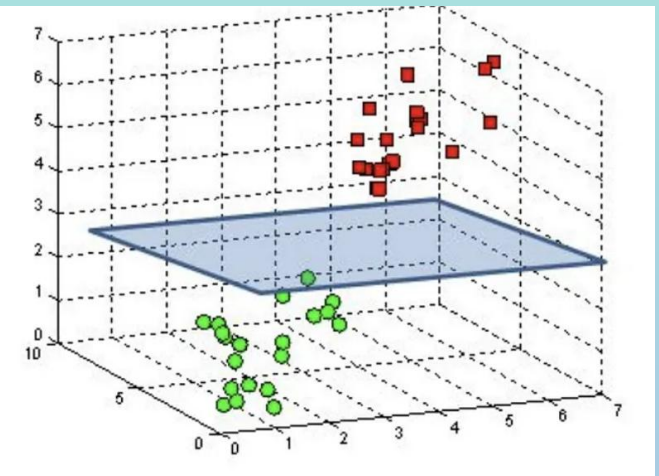


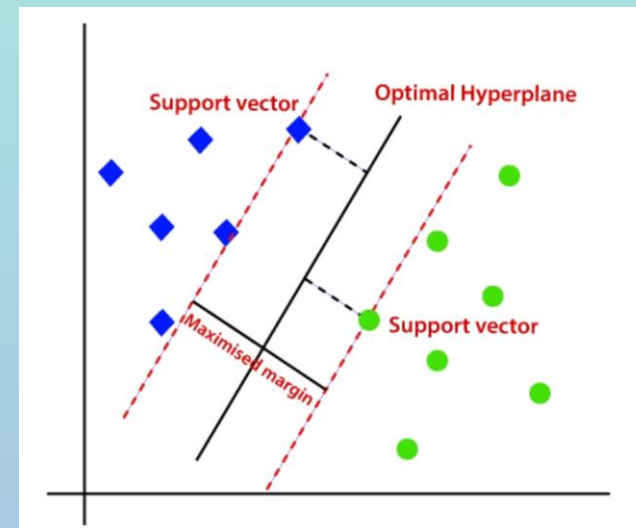
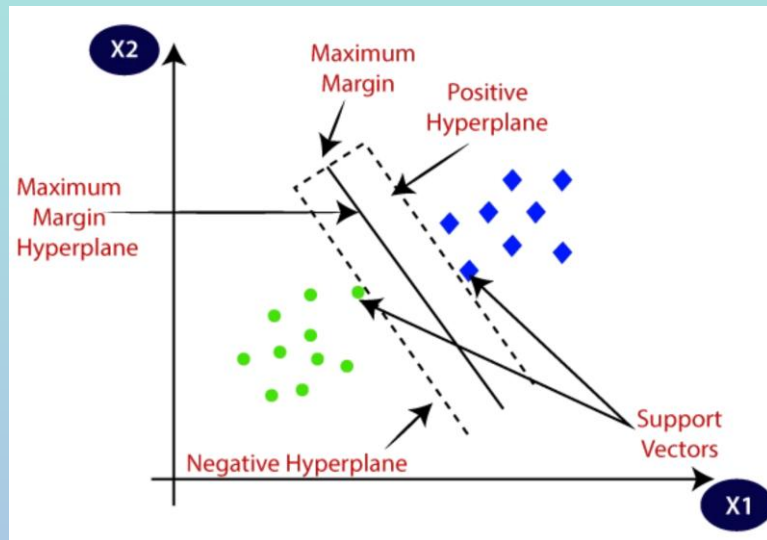
Figure b

- Hyperplane divides a d -dimensional space into two halves.

Supervised Learning

Maximal Margin Classifier

- Identify the best hyperplane that has maximum distance from the both classes.
- **Support Vectors** – Data points that are closest to the hyperplane. Support vectors are used to draw the separating line,
- **Margin** – distance between the hyperplane and the support vectors



Supervised Learning

Maximal Margin Classifier

The hyperplane is defined as

$$W.X + b = 0$$

Where W is the weight vector and b is the bias.

A point (x_1, x_2) *can be considered to be on the hyperplane.*

$$w_1x_1 + w_2x_2 + b = 0$$

$$W.X + b > 0 \text{ for } y = +1$$

$$W.X + b < 0 \text{ for } y = -1$$

$$\text{or simply } y_i(W.X + b) \geq 1$$

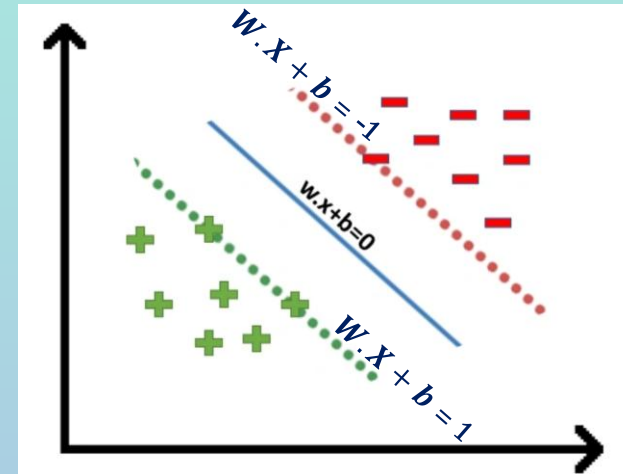
Supervised Learning

Maximal Margin Classifier

The support vector (Positive class) lie on the hyperplane,

$$W \cdot X + b = 1 \quad (\text{Positive side decision boundary})$$

$$W \cdot X + b = -1 \quad (\text{Negative side decision boundary})$$



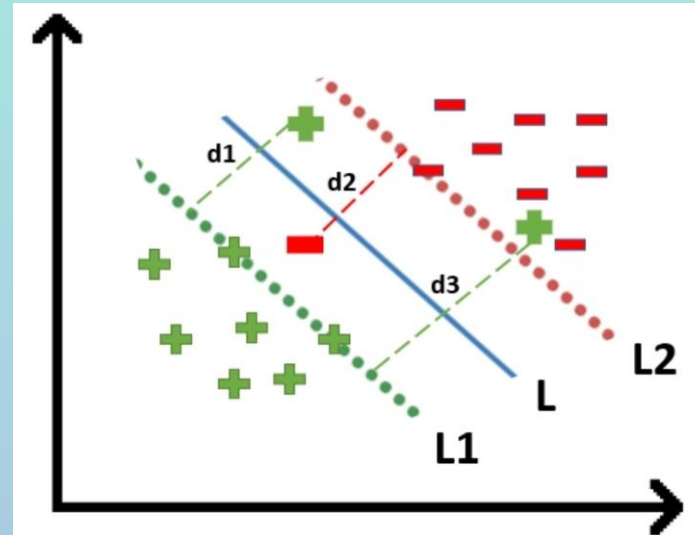
Maximal Margin Classifier works with data that are linearly separable and does not handle noise or outliers.

Supervised Learning

Support Vector Classifier (SVC) – Soft margin

It is a more generalized version, that introduce some flexibility to tolerate some misclassification in the data.

It allows few misclassification, that means it allows few points to be wrongly classified.



Supervised Learning

Support Vector Classifier (SVC) – Soft margin

The optimisation problem for the SVC is to introduce two more terms into the equation, Zeta (ζ) and C

Hence the optimisation equation is

$$\min \frac{1}{2} \|W\|^2 + C \sum_{i=1}^n \zeta_i$$

Subject to $y_i(W \cdot X_i + b) \geq 1 - \zeta_i$

ζ is the variable that measures the degree of misclassification

C is the regularisation parameter that maximise the margin and minimise the classification errors.

Margin error – Error due to allowing data points to lie inside the margin or on the wrong side

Classification error – Error due to data points being misclassified.

Supervised Learning

Support Vector Classifier (SVC) – Soft margin

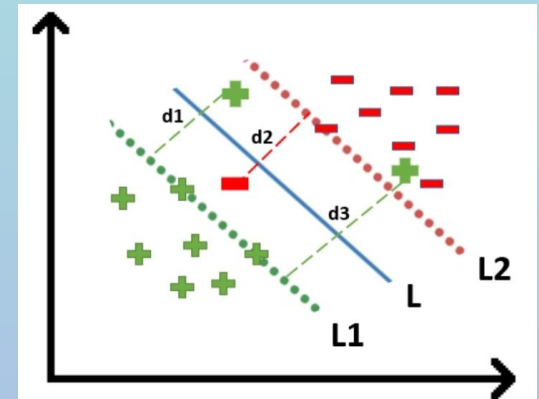
For all the correctly classified points zeta will be equal to 0.

For all incorrectly classified points the zeta is simply the distance of that particular point from its correct hyperplane.

For the wrongly classified green points the value of zeta will be the distance of these points from L1 hyperplane.

For wrongly classified redpoint zeta will be the distance of that point from L2 hyperplane.

$$\text{SVM Error} = \text{Margin Error} + \text{Clasificación Error.}$$



Supervised Learning

Support Vector Classifier (SVC) – Soft margin

SVM Error = Margin Error + Clasificación Error.

Margin Error - violation of the margin constraint.

$$0 \leq \zeta_i \leq 1$$

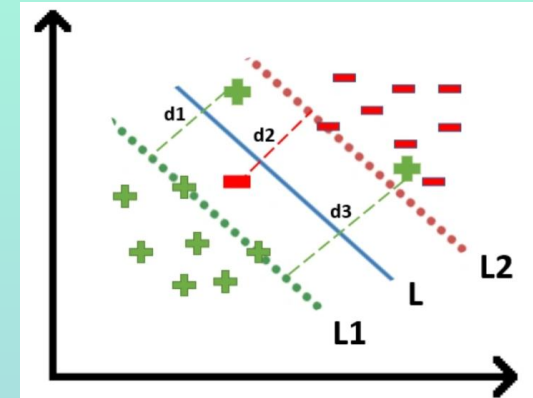
- The data point is on the correct side but on the margin
- It reduces the effective margin does not cause misclassification

Classification Error – when the data point is misclassified. It falls on the wrong side of the decision boundary.

$\zeta_i \geq 1$, The margin violation is too large that the point is incorrectly classified.

The parameter C in the optimisation equation determines how the SVM balances between the two:

- A large C prioritises minimising classification error at the cost of a smaller margin.
- A small C prioritises maximising the margin, even if it allows more margin errors or some misclassified points.



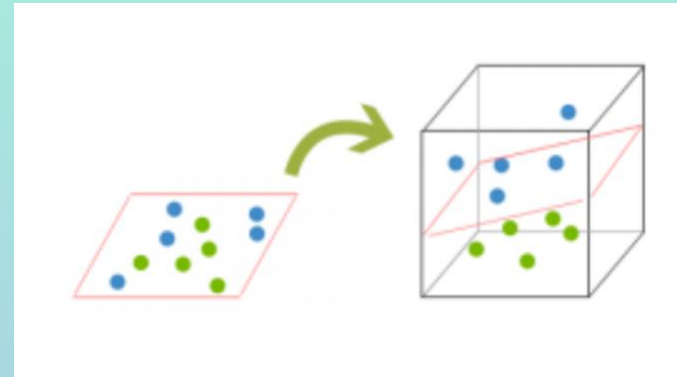
Supervised Learning

Kernel Function

Consider a data set that is not linearly separable



- Mapping to a higher dimension

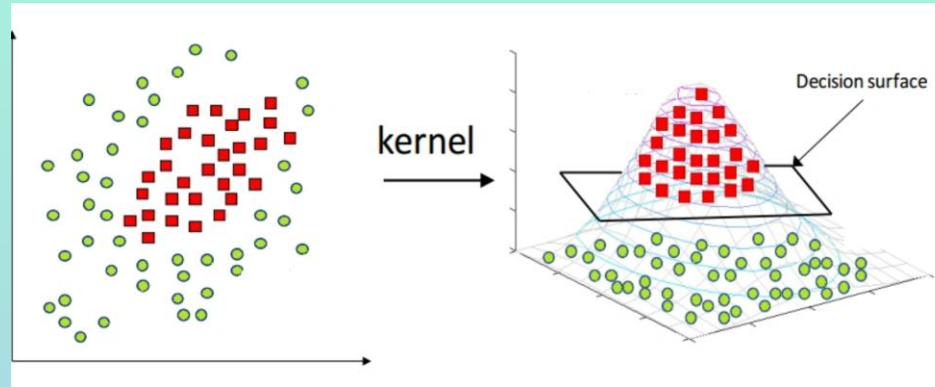


- Data will continue to be mapped into higher dimensions until a hyperplane can be formed to separate the data.

Supervised Learning

Kernel Function

Consider a data set that is not linearly separable



- The function of kernel is to take data as input and transform it into the required form.
- A mathematical function that converts a low-dimensional input space into a higher-dimensional space.
- Example for Kernel functions
- Linear, polynomial, radial basis function (RBF) and sigmoid.

Supervised Learning

Kernel Function

- Linear Kernel Function

$$K(x_i, x_j) = x_i \cdot x_j$$

Linear Kernel A linear kernel can be used as normal dot product any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values.

- Polynomial Kernel Function

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d$$

A more generalised form of the linear kernel.

Where d is the degree of the polynomial. d=1 is similar to the linear transformation. The degree needs to be manually specified in the learning algorithm.

Supervised Learning

Kernel Function

- Radial Basis Function (RBF) or Gaussian Radial Basis

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|x_i - x_j\|^2\right) \text{ where } \gamma = \frac{1}{2\sigma^2}$$

Very popular kernel function. RBF can map an input space in infinite dimensional space.

γ ranges from 0 to 1. A higher value of γ will perfectly fit the training set, which causes overfitting.

Sigmoid

$$K(x_i, x_j) = \tanh(\alpha x_i \cdot x_j + c)$$

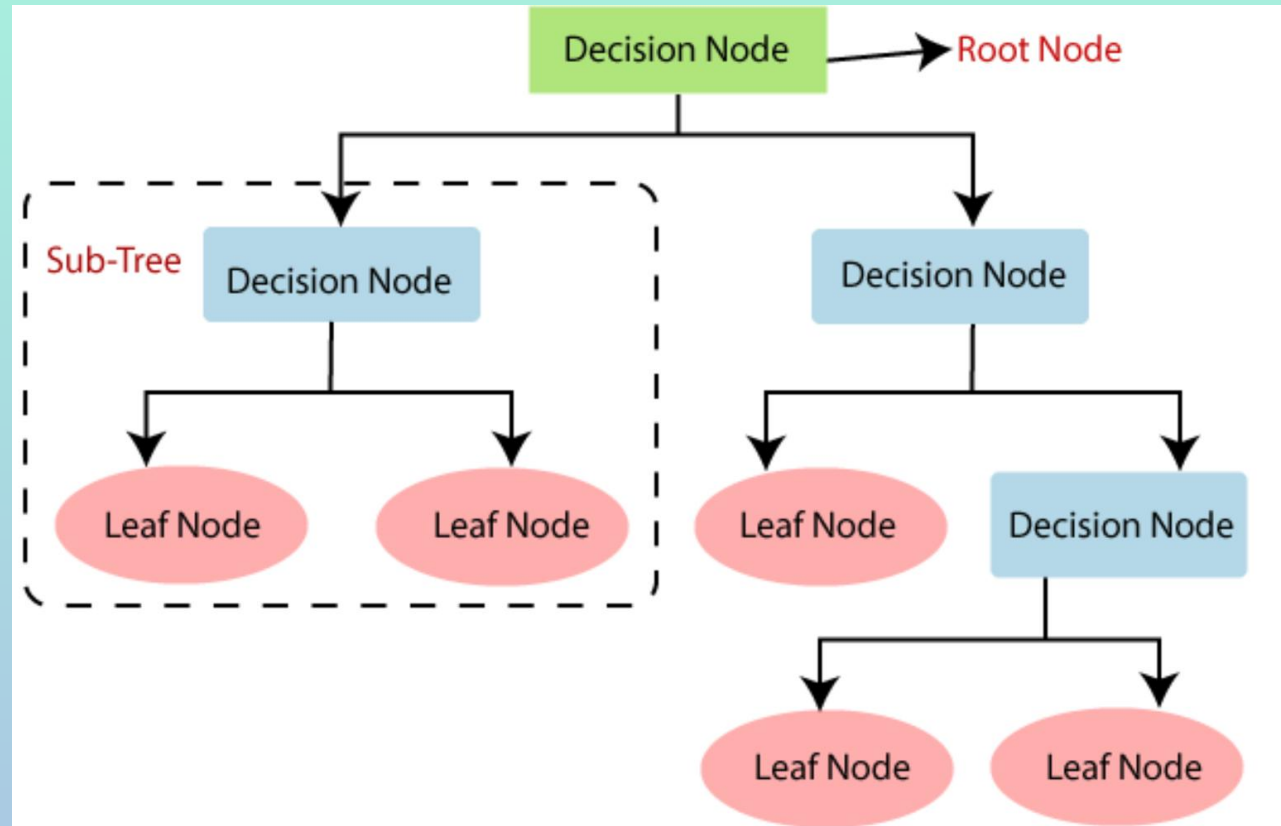
Supervised Learning

Python Demo

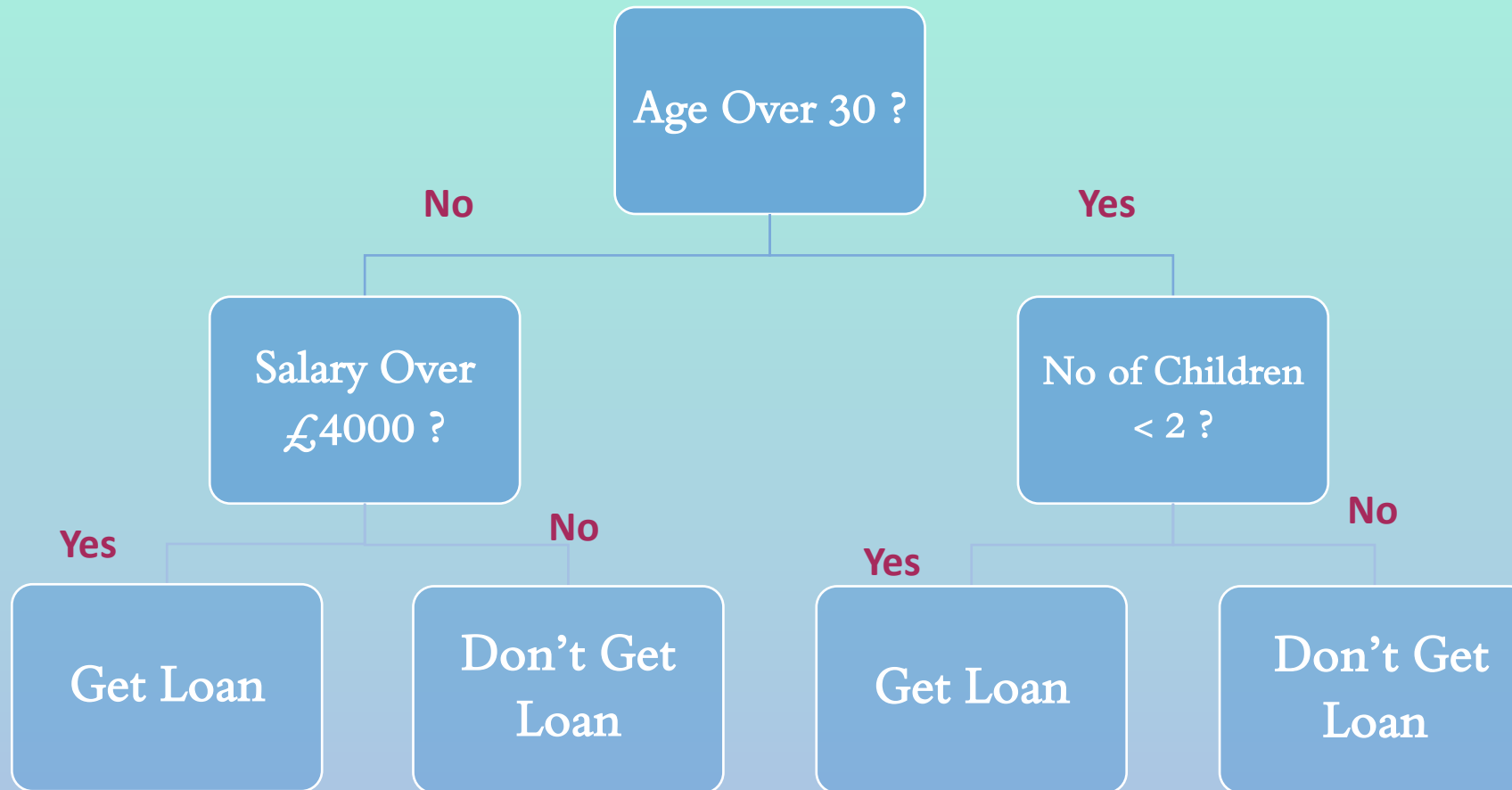
Experiment with different parameters

Supervised Learning - Decision Tree

- General structure of a Decision Tree



Supervised Learning - Decision Tree



Supervised Learning - Decision Tree

- Components

- Root node: the topmost node of the tree where the decision-making process starts based
- Internal node : Represent the decision based on a feature or attribute
- Leaf nodes : Represent the final output or prediction node
- Parent and child node : base node is the parent node and the remaining nodes are child node.
- Branches : A subtree formed by splitting the tree ; outcome of a decision
- Pruning – process of removing unwanted branches from the tree
- Splitting – process of dividing the decision node/ root node into sub-nodes based on the given condition

Supervised Learning - Decision Tree

Attribute/Features – characteristics or property of dataset that is used to make decisions at each node of the tree.

Each attribute corresponds to a question that helps to split the data into subsets.

Type of features

1. Numerical features

Continuous or discrete values (age, salary, temperature)

Numerical features are used to split the tree by comparing against a threshold.. Eg is **Age > 50**

2. Categorical feature

These are discrete, non-numerical values that represent categories or classes. Example is the weather sunny, rainy or cloudy ? Split of categorical features are by checking the membership in a set of values (is the weather sunny ?)

Supervised Learning - Decision Tree

Role of feature in splitting

A feature is used to split the data-based on a specific condition

Example: is **Age** > 50 ? or Is **Salary** high ?

The purpose of feature selection is split the data into homogenous subsets, reducing impurity.

Feature selection – The decision tree algorithm evaluate the multiple features at each node and selects the one with the highest information gain or less impurity.

Supervised Learning - Decision Tree

Attribute Selection Measures (ASM)

How to select the best attribute for the root node and sub nodes.

ASM involves the collection of optimum attributes concerning the source node as well as for the sub-nodes.

The goal of **ASM** is to find the features that split the data into subsets that are homogenous/pure as possible.

Best practices for ASM are:

1. Information gain
2. Gini index

Supervised Learning - Decision Tree

Information gain- Calculates the amount of information that is provided by a feature regarding the class.

Information Gain = Entropy (S) – [(Weighted Average)*Entropy (Every Feature)]

Entropy – is the measure of impurity or disorder or uncertainty;

Entropy values : range from 0 to 1; less the entropy value better the outcome;

Entropy controls how a decision tree decides to split the data.

Supervised Learning - Decision Tree

How to calculate entropy ?

$$\text{Entropy}(s) = -P(\text{Yes})\log_2 P(\text{Yes}) - P(\text{No})\log_2 P(\text{No})$$

where $P(\text{Yes}) \rightarrow$ probability/Chance of getting yes

$P(\text{No}) \rightarrow$ probability/Chance of getting No

Supervised Learning - Decision Tree

Gini Index – Measures the impurity or purity used during the creation of a decision tree algorithm.

$$Gini(D) = 1 - \sum_{i=1}^k p_i^2$$

Where p_i is the probability of choosing a sample of class i .

K is the number of classes

A lower Gini Index represents the better split

Supervised Learning - Decision Tree

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Supervised Learning - Decision Tree

Example

Dataset: Weather and Decision to Play

Weather	Windy	Play?
Sunny	No	No
Sunny	Yes	No
Overcast	No	Yes
Rainy	No	Yes
Rainy	Yes	No
Rainy	No	Yes
Overcast	Yes	Yes
Sunny	No	Yes

- Feature 1 : Weather (Sunny, Overcast, Rainy)
- Feature 2: Windy (Yes, No)
- Target: Play? (Yes, No)

Supervised Learning - Decision Tree

Example for calculating information gain

Number of instances for Yes or No for the Target variable, Play.

Yes = 5

No = 3

Step 1:

Entropy for the whole dataset, S

$$\text{Entropy}(S) = -P(\text{Yes})\log_2 P(\text{Yes}) - P(\text{No})\log_2 P(\text{No})$$

$$= -5/8\log_2(5/8) - (3/8)\log_2(3/8)$$

$$= -(0.625 \log_2 0.625 + 0.375 \log_2 0.375)$$

$$= -(0.625 \times -0.678 + 0.375 \times -1.415) = 0.954$$

Supervised Learning - Decision Tree

Step 2:

Calculate the entropy for Weather

Sunny; Overcast; Rainy

Sunny = 3 instances with 2 No and 1 Yes (entropy =0.918)

Overcast = 2 instances ; 2 Yes (entropy =0)

Rainy = 3 instances with 2 No and 1 Yes ; (entropy =0.918)

Weighted entropy : $\frac{3}{8} \times 0.918 + \frac{2}{8} \times 0 + \frac{3}{8} \times 0.918 = 0.688$

Step 3: entropy (s) – entropy (weather)

= 0.954 - 0.688 = **0.266**

Supervised Learning - Decision Tree

Example Gini Index

Number of instances for Yes or No for the Target variable, Play.

Yes = 5

No = 3

Step 1:

$$\begin{aligned} \textit{Gini}(S) &= 1 - \left(\left(\frac{5^2}{8} \right) + \left(\frac{3^2}{8} \right) \right) \\ &= 1 - (0.3906 + 0.1406) = 1 - 0.5312 = 0.4688 \end{aligned}$$

Supervised Learning - Decision Tree

Example Gini Index

Step 2 : Calculate Gini Index for Splitting on "Weather"

Sunny : 3 instances, 2 "No", 1 "Yes" (Gini = 0.444).

Overcast: 2 instances, 2 "Yes" (Gini = 0).

Rainy: 3 instances, 2 "Yes", 1 "No" (Gini = 0.444).

Calculate the weighted Gini Index

$$= 3/8 \times 0.444 + 2/8 \times 0 + 3/8 \times 0.444 = 0.333$$

Supervised Learning - Decision Tree

Example interpretation

Information Gain for splitting on "Weather" is **0.266**. This shows that splitting on this feature reduces uncertainty (entropy), but the reduction is moderate.

Gini Index after the split is **0.333**, indicating that the groups formed by this split are somewhat pure, though not entirely.

Supervised Learning - Decision Tree

Example interpretation

Information Gain for splitting on "Weather" is **0.266**. This shows that splitting on this feature reduces uncertainty (entropy), but the reduction is moderate.

Gini Index after the split is **0.333**, indicating that the groups formed by this split are somewhat pure, though not entirely.

Extra reading

[Scikit-learn SVM Tutorial with Python \(Support Vector Machines\) | DataCamp](#)

[What Is Support Vector Machine? | IBM](#)

[Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science](#)

[Support Vector Machine Algorithm \(SVM\), SVM Formula](#)

[Decision Tree Algorithm in Machine Learning – Javatpoint](#)

Reference

Géron, Aurélien. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems*. Third edition. Sebastapol, CA: O'Reilly Media, Inc., 2023. Print.