ARITIFICAL INTELLIGENCE A MODERN APPROACH from p.1200

MLU

SML

UML

data

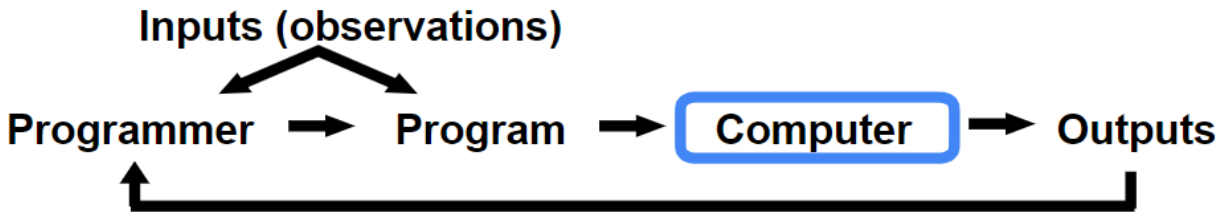
1.1 ML Landscape link, link

1.1.1 What is Machine Learning?

Machine learning is a branch of computer science and a field of study within artificial intelligence. It emphasizes the creation and exploration of statistical algorithms capable of learning from data and extrapolating to new data sets. ML enables systems to execute tasks without direct programming instructions. It focuses on developing models and algorithms that empower computers to enhance their performance based on data without explicit programming. The core principle of machine learning is to facilitate automated learning and improvement through experiential data analysis.

1.1.2 Why to use Machine Learning?

Traditional programming relies heavily on developers defining explicit rules, which can become brittle and difficult to manage, especially as complexity increases. Machine learning excels at uncovering hidden patterns and relationships within data that humans might not even think to program. This data-driven approach often leads to more robust and accurate solutions.



Moreover, in real-world scenarios, things rarely stay the same. Customer preferences shift, language patterns evolve, and new trends emerge. Machine learning models, with the right design, can adapt alongside these changes. This makes them significantly more resilient than rule-based systems that need constant manual updates.

Furthermore, Machine learning offers a pathway to solve problems that defy traditional algorithmic approaches. Consider tasks like recommending the perfect movie, accurately translating between languages, or driving a car autonomously. These were once considered incredibly difficult,

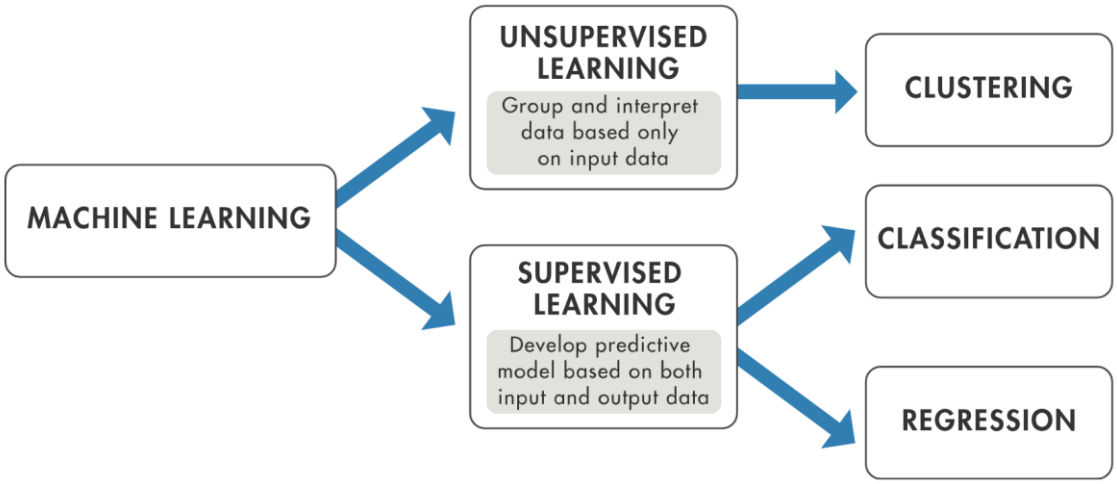
if not impossible, with conventional programming, but machine learning is steadily making progress in these domains.

Nevertheless, Machine learning 'learns by doing.' By feeding a model a large set of examples, it can discover the underlying trends and features that differentiate spam from legitimate emails, medical conditions from X-rays, or promising stock picks from financial data. This can be far more powerful than trying to manually pinpoint all relevant factors.

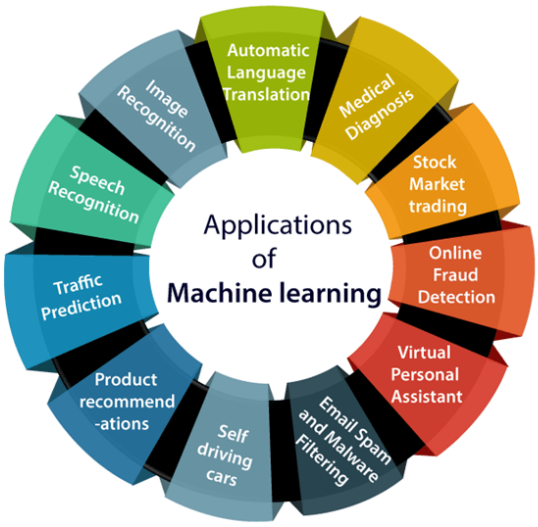
Additionally, many machine learning models grow smarter with experience. As they are exposed to more data, they can refine their internal representations and decision-making, allowing for continuous improvement without needing a developer to make code changes.

1.1.3 How it works?

Machine learning uses two types of techniques: supervised learning, which trains a model on known input and output data so that it can predict future outputs, and unsupervised learning, which finds hidden patterns or intrinsic structures in input data.



1.1.4 Applications of Machine Learning



1.2 Categories of Machine Learning

1.2.1 Supervised Learning

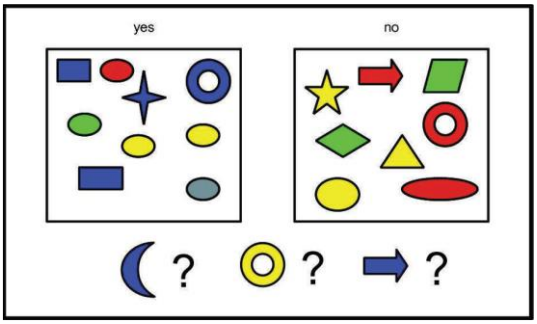
In predictive or supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels. The goal is to learn a mapping from inputs x to outputs y, given a labeled set of input-output pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. Here D is called the training set, and N is the number of training examples.

In the simplest setting, each training input x_i is a D-dimensional vector of numbers, representing, say, the height and weight of a person. These are called features, attributes or covariates. In general, however, x_i could be a complex structured object, such as an image, a sentence, an email message, a time series, a molecular shape, a graph, etc.

Similarly, the form of the output or response variable can in principle be anything, but most methods assume that y_i is a categorical or nominal variable from some finite set, $y_i \in \{1, \ldots, C\}$ (such as male or female), or that y_i is a real-valued scalar (such as income level). When y_i is categorical, the problem is known as classification or pattern recognition, and when y_i is real-valued, the problem is known as regression.

1.2.1.1 Classification

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups.



Here the goal is to learn a mapping from inputs x to outputs y, where $y \in \{1, \ldots, C\}$, with C being the number of classes. If C = 2, this is called binary classification (in which case we often assume $y \in \{0, 1\}$); if C > 2, this is called multiclass classification.

We assume $y = f(\mathbf{x})$ for some unknown function f, and the goal of learning is to estimate the function f given a labeled training set, and then to make predictions using $\hat{y} = \hat{f}(\mathbf{x})$.

To handle ambiguous cases, it is desirable to return a probability. The probability distribution over potential labels, considering the input vector x and training set D, is denoted as p(y|x, D), typically representing a vector of length C. From this probabilistic output, we can derive our best estimate of the true label using the argmax function to find

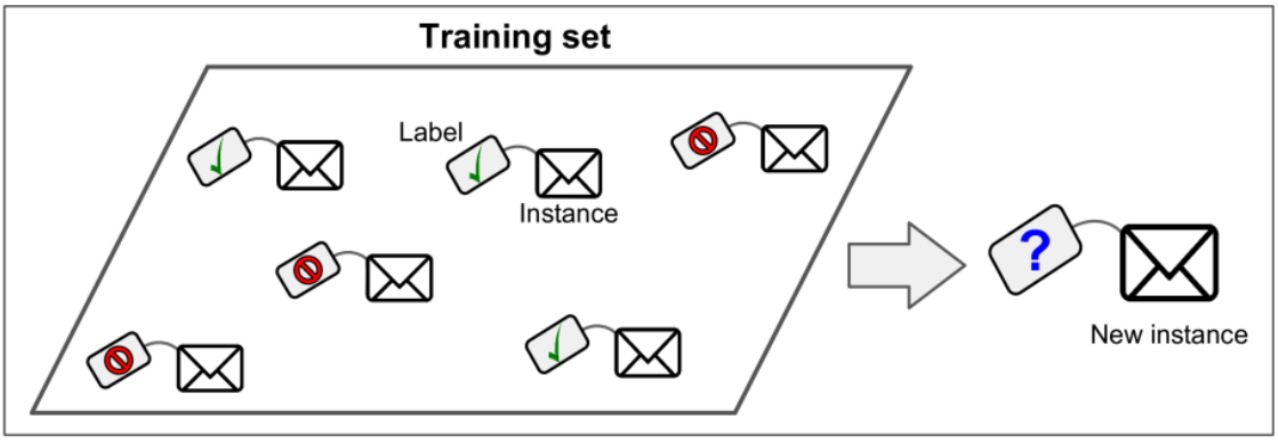
$$\hat{y} = \hat{f}(\mathbf{x}) = \underset{c=1}{\overset{C}{\operatorname{argmax}}}\, p(y = c|\mathbf{x}, \mathcal{D})$$

This corresponds to identifying the most probable class label, known as the mode of the distribution p(y|x, D).

Classification Algorithms can be further divided into the Mainly two category:

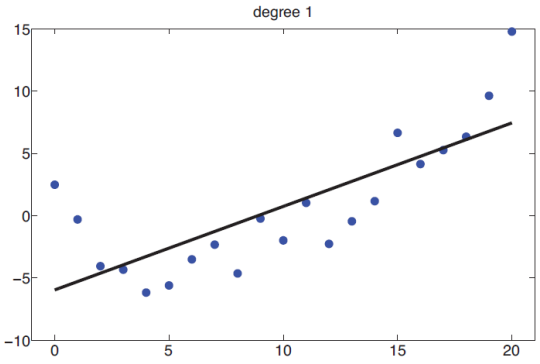
- **Linear Models**
  - Logistic Regression
  - Support Vector Machines
- **Non-linear Models**
  - K-Nearest Neighbours
  - Kernel SVM
  - Naïve Bayes
  - Decision Tree Classification
  - Random Forest Classification

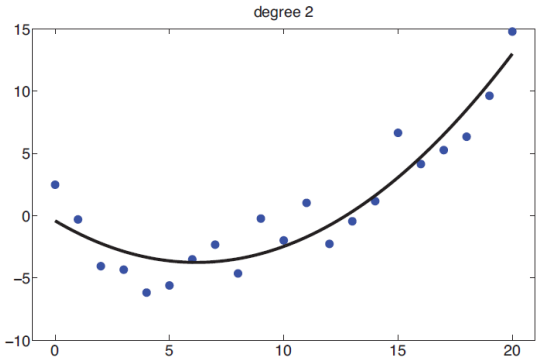
An example of a classification task can be considered the spam classifier:



1.2.1.2 Regression

Regression is just like classification except the response variable is continuous. We have a single real-valued input $x_i \in \mathrm{R}$, and a single real-valued response $y_i \in \mathrm{R}$. We consider fitting two models to the data: a straight line and a quadratic function.





The main regression models are:

- Linear Regression
- Logistic Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- Ridge Regression
- Lasso Regression

As a regression example consider the prediciton of housing prices.

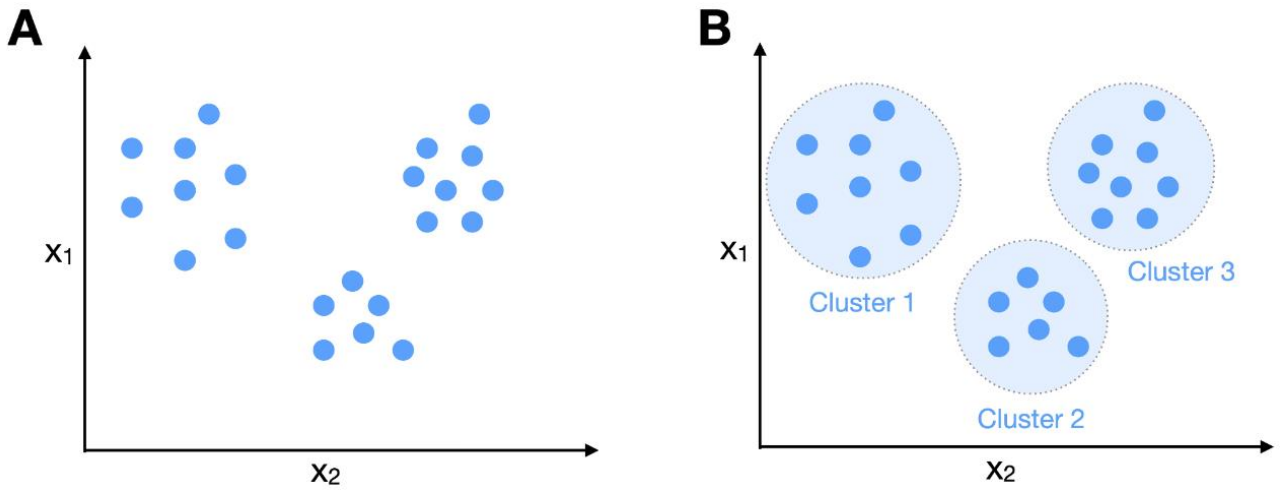
1.2.2 Unsupervised Learning

In the descriptive or unsupervised learning approach, we are only given inputs, meaning the training data is unlabeled, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$, and the goal is to find "interesting patterns" in the data. This is sometimes called knowledge discovery.

Unlike supervised learning, we are not told what the desired output is for each input. Instead, we will formalize our task as one of density estimation, that is, we want to build models of the form $p(x_i|\theta)$.

1.2.2.1 Clustering

Clustering is a way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group. It is done by finding some similar patterns in the unlabeled dataset such as shape, size, color, behavior, etc., and by dividing them as per the presence and absence of those similar patterns.



One of the most popular clustering algorithms is K-Means.

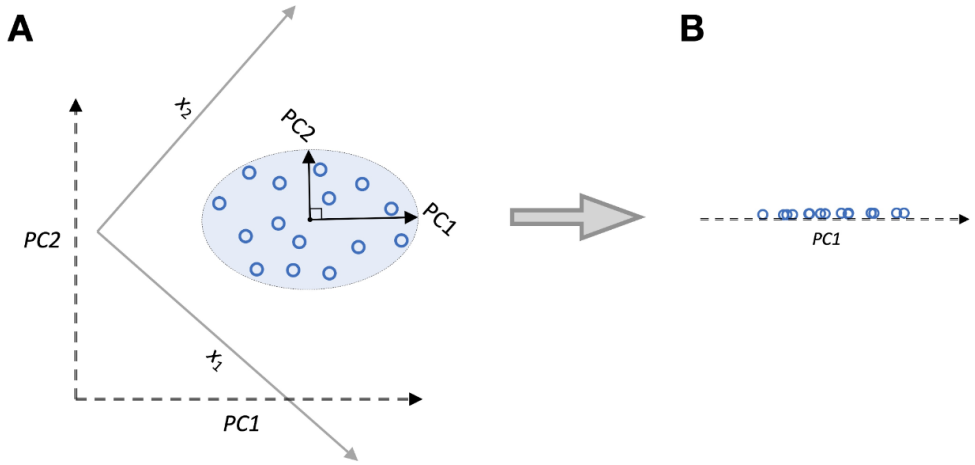
For a clustering example consider the customer segmentation in marketing, where the clusters are based on purchasing behavior of the customers.

1.2.2.2 Dimensionality Reduction

Many Machine Learning problems involve thousands or even millions of features for each training instance. Not only do all these features make training extremely slow, but they can also make it much harder to find a good solution. This problem is often referred to as the curse of dimensionality. Fortunately, in real-world problems, it is often possible to reduce the number of features considerably, turning an intractable problem into a tractable one.

The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction. ***It is a way***

of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information.



Reducing dimensionality does cause some information loss, so even though it will speed up training, it may make the system perform slightly worse.

1.2.2.3 Anomaly Detection

Anomaly detection is a process of finding those rare items, data points, events, or observations that make suspicions by being different from the rest data points or observations. Anomaly detection is also known as outlier detection.

1.2.3 Reinforcement Learning

Reinforcement learning trains software to make decisions to achieve the most optimal results. It mimics the trial-and-error learning process that humans use to achieve their goals. RL algorithms use a reward-and-punishment paradigm as they process data. They learn from the feedback of each action and self-discover the best processing paths to achieve final outcomes.

Reinforcement learning is based on the Markov decision process, a mathematical modeling of decision-making that uses discrete time steps. The probability distribution that governs the likelihood of an agent transitioning from one state to another when it takes a particular action in the environment.

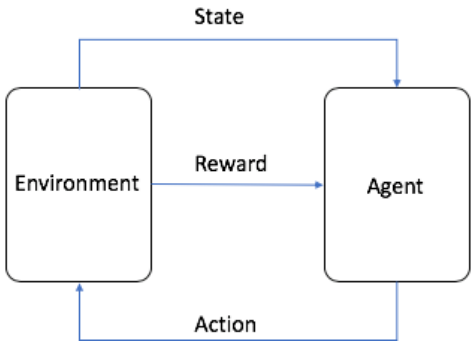
$$p(s'|s,a) = Pr(S_{t+1} = s'|S_t = s, A_t = a)$$

Where:

- a set of environment and agent states, S;
- a set of actions, A, of the agent;

At every step, the agent takes a new action that results in a new environment state. Similarly, the current state is attributed to the sequence of previous actions.

Through trial and error in moving through the environment, the agent builds a set of if-then rules or policies. The policies help it decide which action to take next for optimal cumulative reward. The agent must also choose between further environment exploration to learn new state-action rewards or select known high-reward actions from a given state. This is called the *exploration-exploitation trade-off*.



An example of reinforcement learning is teaching a computer program to play a video game. The program learns by trying different actions, receiving points for good moves and losing points for mistakes.

1.2.4 Semi-Supervised Learning

Semi-supervised learning is a branch combines supervised and unsupervised learning by using both labeled and unlabeled data to train models for classification and regression tasks.

Semi-supervised learning uses pseudo labeling to train the model with less labeled training data than supervised learning. The process can combine various neural network models and training ways.

The whole working of semi-supervised learning is explained in the below points:

- Firstly, it trains the model with less amount of training data similar to the supervised learning models. The training continues until the model gives accurate results.
- The algorithms use the unlabeled dataset with pseudo labels in the next step, and now the result may not be accurate.
- Now, the labels from labeled training data and pseudo labels data are linked together.
- The input data in labeled training data and unlabeled training data are also linked.

- In the end, again train the model with the new combined input as did in the first step. It will reduce errors and improve the accuracy of the model.

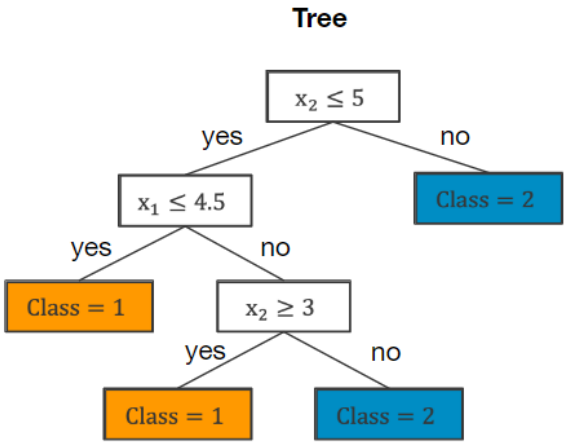
Text classification can be considered as an example of semi-supervised learning. In text classification, the goal is to classify a given text into one or more predefined categories.

3. Supervised learning, Decision Trees

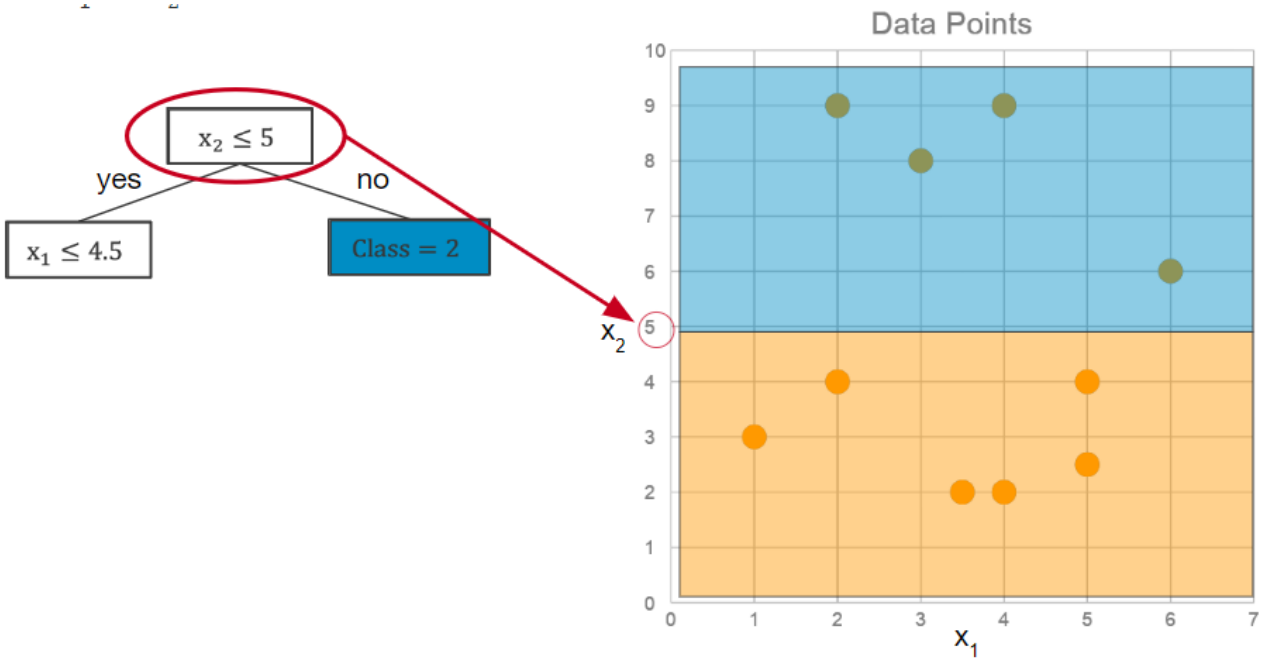
Geron p.175-185, link, link2, link3, link

3.1 What are decision trees

Classification and regression trees or Decision trees are defined by recursively partitioning the input space, and defining a local model in each resulting region of input space. It is a series of yes/no questions about your input that you ask in sequence.

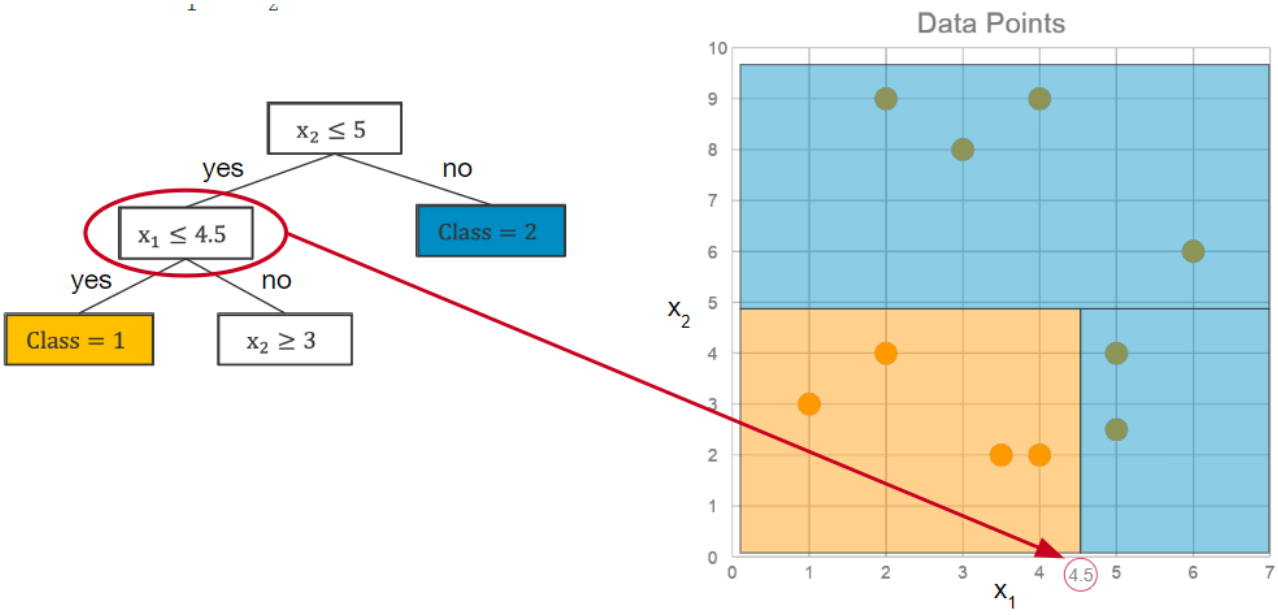


Which feature x_1 or x_2 to split first to best separate class 1 from class 2?

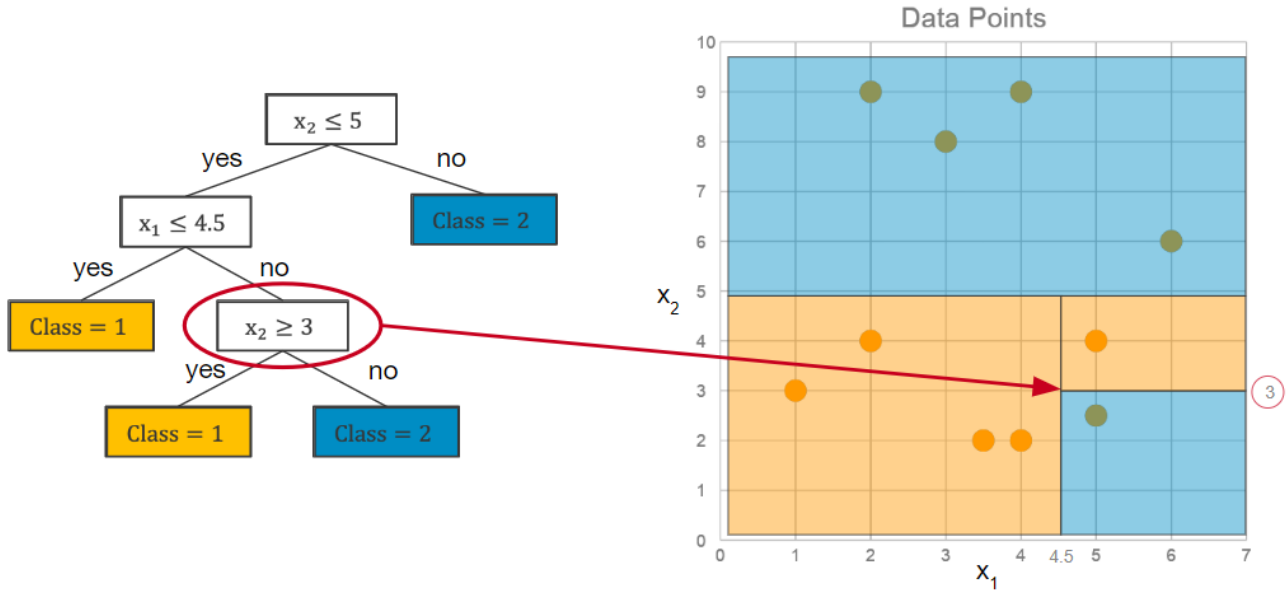


Which feature x_1 or x_2 to split second to best separate class 1 from class 2?

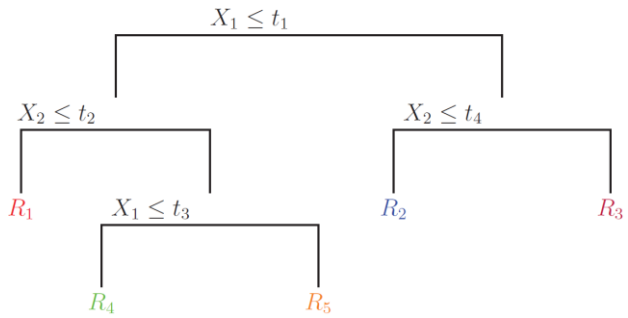
A.K.



Which feature x_1 or x_2 to split third to best separate class 1 from class 2?



A general example:



The split function that chooses the best feature, and the best value for that feature:

A.K.

$$(j^*, t^*) = \arg \min_{j \in \{1,\ldots,D\}} \min_{t \in \mathcal{T}_j} \text{cost}(\{\mathbf{x}_i, y_i : x_{ij} \leq t\}) + \text{cost}(\{\mathbf{x}_i, y_i : x_{ij} > t\})$$

Traditionally the decision trees are made by splitting on a single comparison of a single input feature. The splitting process ends with a stopping criteria:

- Once you have reached a maximum depth
- Once you have reached a maximum number of "leaves" (the regions in which you assign a particular decision)
- Once there are too few data points in a particular leaf
- Once the leaf has met some desired level of purity

DT

3.2 Impurity Functions

Slides p.23, Murphy p.547, Geron p.180

3.3 Basic Properties of Decision Trees

Slides p.53

3.4 Basic Reuglarization of Decision Trees

Slides p.55, Geron p.181, video, RF, link

4. ML Model Evaluation

Murphy p. 217, colab, Precision and Recall, Geron p.88, Accuracy, Roc Curve, link, link2, kaggle, regression, mlu

5. Regression Models

link, math, Murphy p.21, Geron p.142, link, link2, link3, slides, link4, link5, link6 , link

6. Logistic Regression

video, Murphy p.245, Geron p.142, link, link, link, mlu

7. KNN

link, link2, link3, link, link, link

8. SVM <=

slides, link, link2, video, video2, video3, link, slides, video4, video5, link4, mit, link

9. Naive Bayes

link, link2, link3, slides, Murphy p.82,311, video, video2, video3, link, link, math, wiki, link

10. Unsupervised learning, K-Means <<=

link, link2, link3, link4, video, video2, video3, Geron p.238, Murphy p.352, link, link, link, link, visual, math, link

11. Neural Networks <<<=

Geron p.279, murphy p.563, 999, video, video2, video3, video4, MATH, video5, link, link, link, link, videos, simulator, video, link, link, link

EDA

RL

Train_Test_Split

CV

SGD

$$\sum_{x \in C_i} X$$

slides, link, link2, video, video2, video3, link, slides, video4, video5, link4

$\mathbf{x}_i^{\mathbf{T}} \beta + \varepsilon_i$

$\frac{\partial l}{\partial \beta}$