Unsupervised learning: clustering

Supervised learning: decision-trees

25.11.2024

Outline

- Hour 1
  - Brief review of PCA
  - Unsupervised learning: clustering

- Hour 2: back to supervised learning
  - Decision-trees
    - For regression
    - For classification

PCA - example

$x^i \in \mathbb{R}^d$

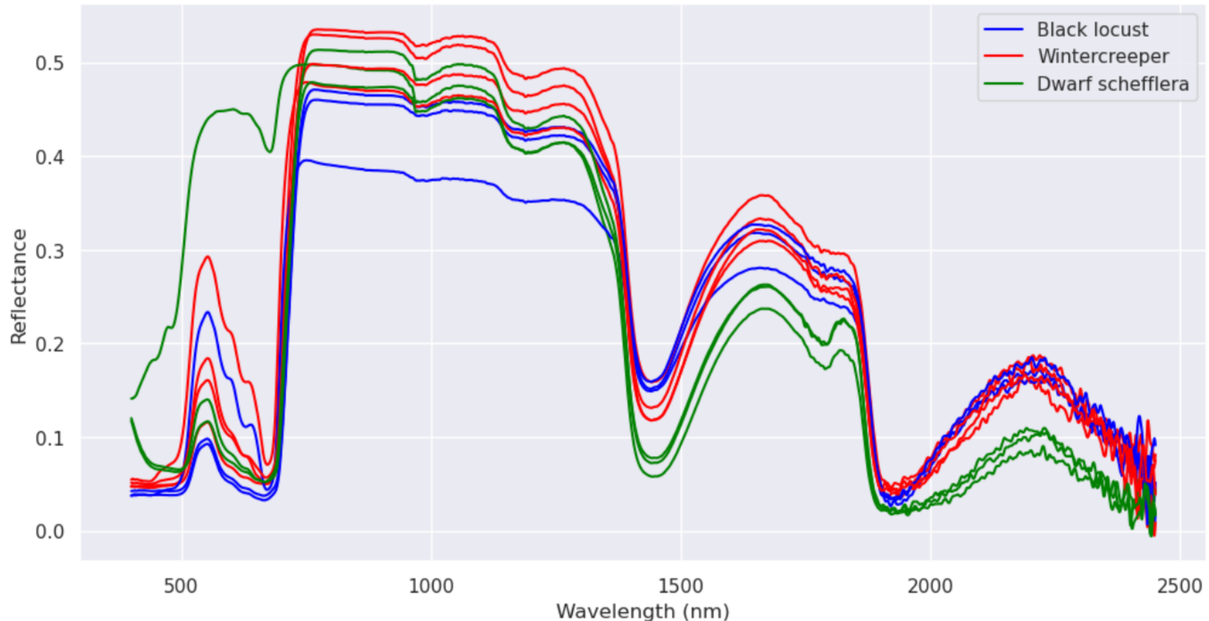
- **Dataset:** consider an unlabelled normalised data set containing N sample points and d features. $\{x^i\}_{i=1}^N$
  - How many eigenvalues would the data covariance matrix have? $X \in \mathbb{R}^{N \times d}$

    $C = X^T X \in \mathbb{R}^{d \times d}$, d eigenvalues $\lambda_i \in \mathbb{R}$ $i = 1, 2, \cdots, d$.
  - Suppose we perform PCA and find approximate the data based on its 3 principal components. Let $\Theta \in \mathbb{R}^{d \times r}$ be the associated eigenvectors. What is the projection of a given data point $x^i \in \mathbb{R}^d$ on the space spanned by these eigenvectors?

$\Theta = [\theta_1 | \theta_2 | \cdots | \theta_r] \in \mathbb{R}^{d \times r}$

$r = 3$

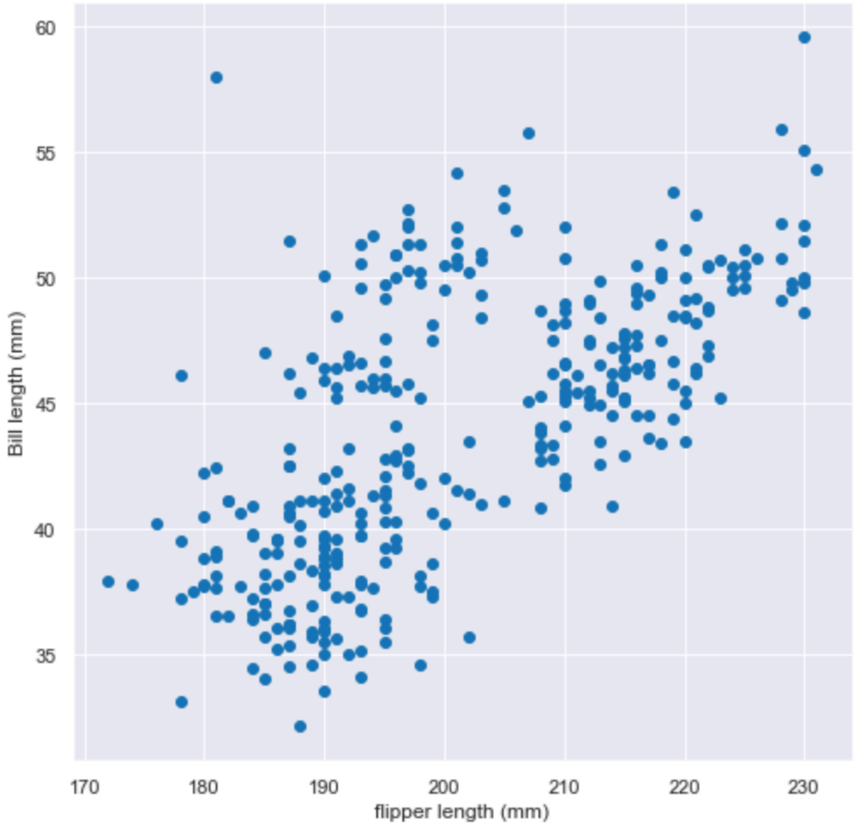
$\hat{x}^i = \Theta \Theta^T x^i \in \mathbb{R}^d$



Clustering

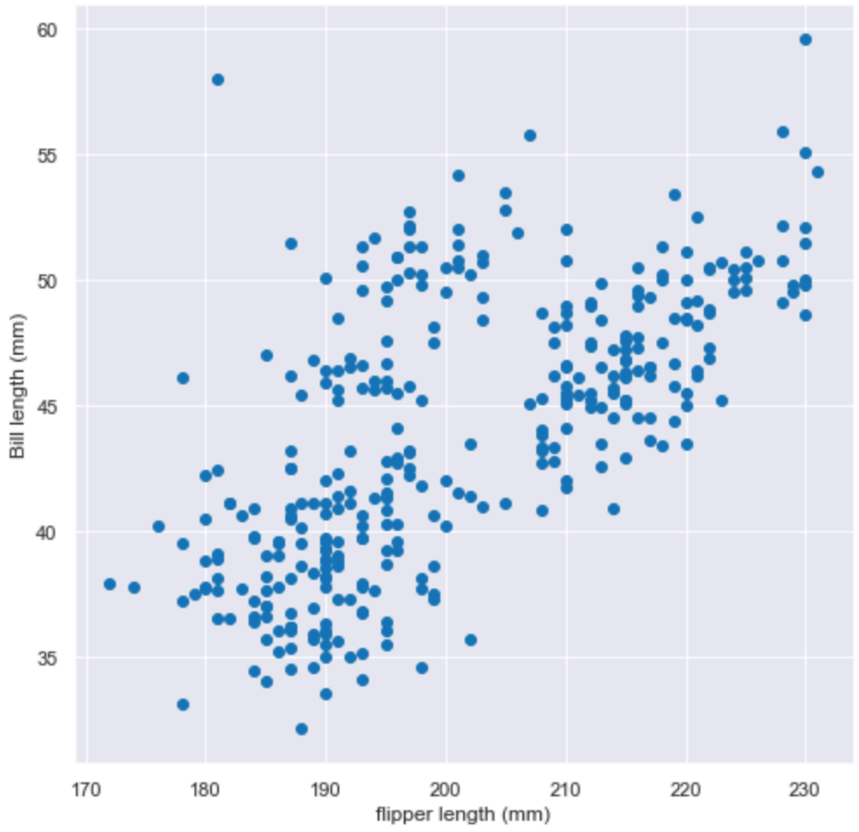
k-means
Example

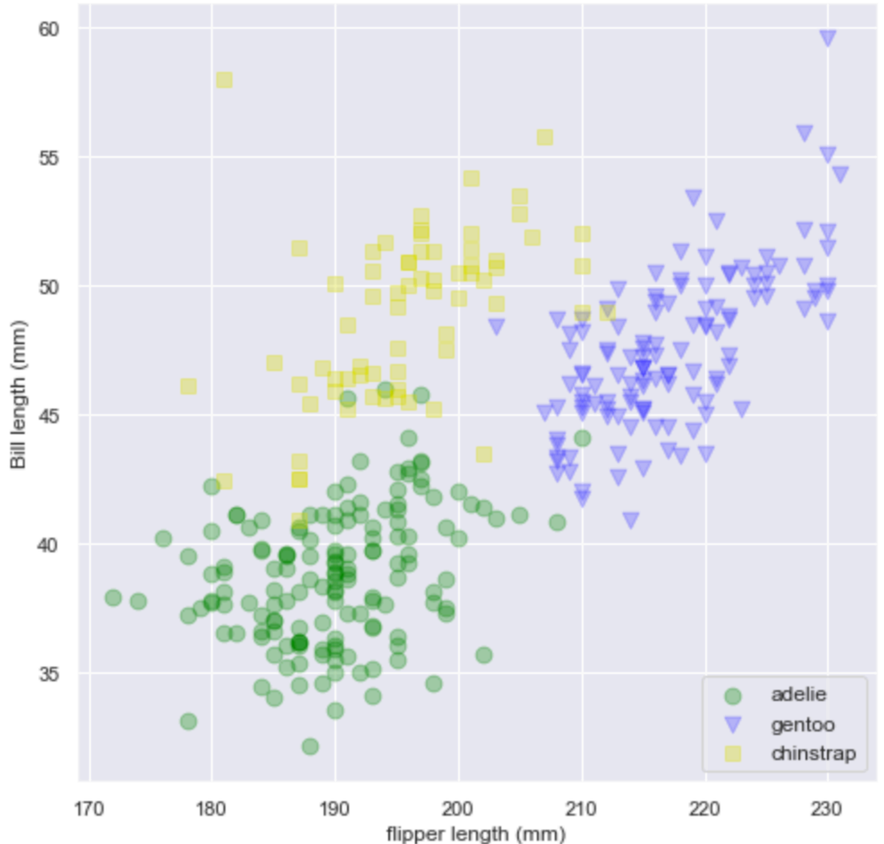
- Palmer Penguins dataset
  - Features: Flipper length and Bill length
  - Can we identify k species based on this data?



k-means
Example

- Note that when we apply clustering, we don’t have the labels. Our goal is to be able to best way to cluster the data (a bit vague but we will formalise one way to do this)





k-means approach to clustering

How to cluster data without labels?

Try to find similarity between groups of points

k-means: Group points based on their proximity
(in terms of distance in the feature space)

- Given a set of unlabelled input samples, group the samples into k clusters ($k \in \mathbb{N}$)
- **k-Means idea:**
  - Identify k cluster of data points given N samples**.**
  - Find prototype points $\mu^1, \mu^2, \ldots, \mu^k \in \mathbb{R}^d$ representing the center of each cluster and add the data points to the nearest cluster

k-means
Preliminaries

Let $\{x^i\}_{i=1}^N \subset \mathbb{R}^d$ be our data set

- A single representative point for data: want to find $\mu \in \mathbb{R}^d$ that represents our data.

- Example: suppose we want to have one representative point
  - we use a mean-squared $J(\mu) = \frac{1}{N} \sum_{i=1}^N \|x^i - \mu\|_2^2 \Rightarrow \min_\mu J(\mu) \Rightarrow \hat{\mu} = \frac{\sum_{i=1}^N x^i}{N}$
  - we use an absolute value loss $J(\mu) = \frac{1}{N} \sum_{i=1}^N |x^i - \mu| \Rightarrow \hat{\mu} = \text{median}(\{x^i\}_{i=1}^N)$

k-Means
Approach

$\{x^i\}_{i=1}^N$

- Choose k clusters to represent data $\{w^1, w^2, \dots, w^K\} \subset \mathbb{R}^d, \quad w^j \in \mathbb{R}^d$

  (How do we find $\{w^j\}_{j=1}^K$?

- Determining the cluster a single point x^i belongs to $\min_{j=1,2,\dots,K} \| x^i - w^j \|_2^2$

  $c(i)$: cluster x^i belongs to. $c(i) = \underset{j=1,2,\dots,K}{\operatorname{argmin}} \| x^i - w^j \|_2^2$

- Determining the cluster centres to minimize the distance of each point to its assigned cluster

$$\min_{\{w^j\}_{j=1}^K} \frac{1}{N} \sum_{i=1}^N \| x^i - w^{c(i)} \|_2^2 = \min_{\{w^j\}_{j=1}^K} \frac{1}{N} \sum_{i=1}^N \min_{j=1,2,\dots,K} \| x^i - w^j \|_2^2$$

k-Means heuristic algorithm

Loss is non-convex ⇒ heuristic

Algorithm -

1. Initialize $\{\mu^1, \mu^2, \ldots, \mu^k\}$ (e.g., randomly)
2. While not converged
   1. Assign each point x^i to the nearest center (see previous slide)
   2. Update each center μ^j based on the points assigned to it $\mu^j = \frac{1}{N_j} \sum_{i \in ind_j} x^i$

ind_j : indices of points assigned to cluster j.

- **Step 2.1:** For each point x^i, compute the Euclidean distance to every center $\{\mu^1, \mu^2, \ldots, \mu^k\}$
  - Find the smallest distance $\min_j \| x^i - \mu^j \|_2^2$ for each data point
  - The point is said to be assigned to the corresponding cluster (note that each point is assigned to a single cluster)
- **Step 2.2:**
  - Recompute each center μ^j as the mean of the points that were assigned to it

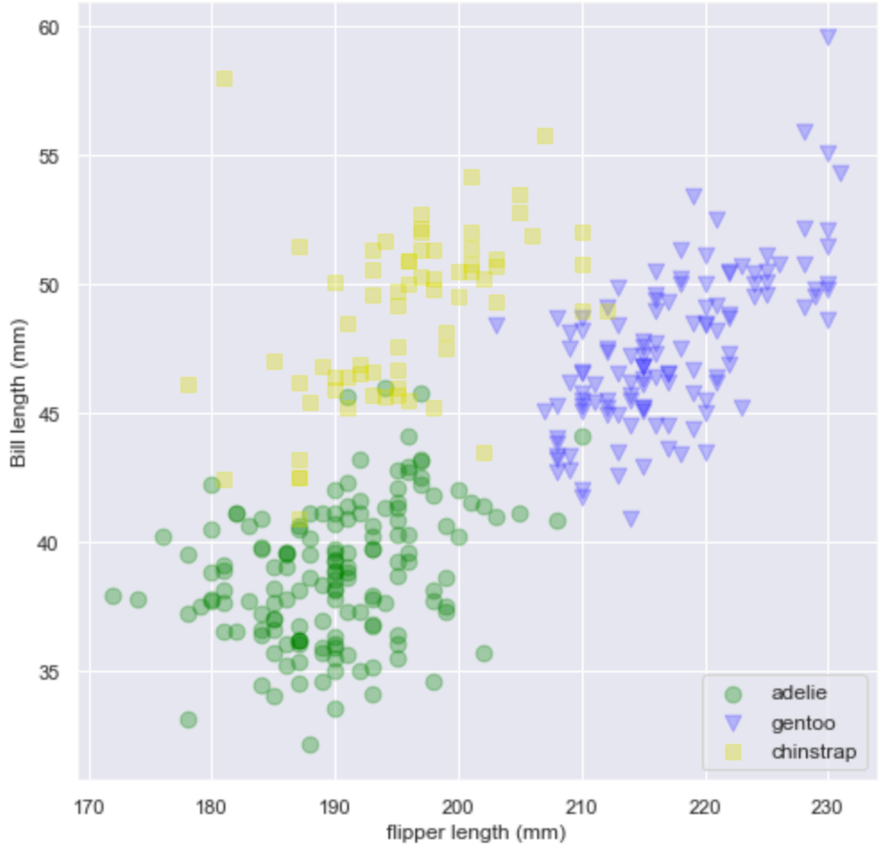
k-means
Algorithm - Convergence

- **Step 2** is repeated while k-means has not converged
  - What criteria to stop iterating?
  - Fixed number of iterations? It's arbitrary and a too small number can lead to bad results
  - The difference in assignments or center locations between two iterations can be used as criteria to stop the algorithm

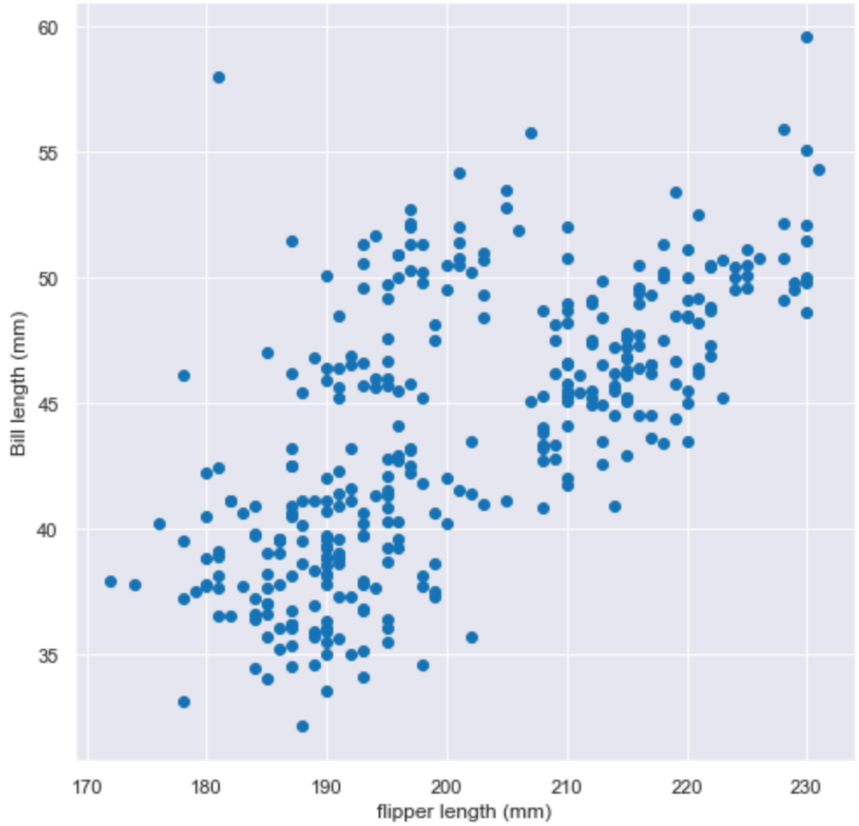
- **k-means does not** always converge to the **best solution**
  - Non-convex optimization problem

k-means
Example

- Use the Palmer Penguins dataset
  - With Flipper length against Bill length



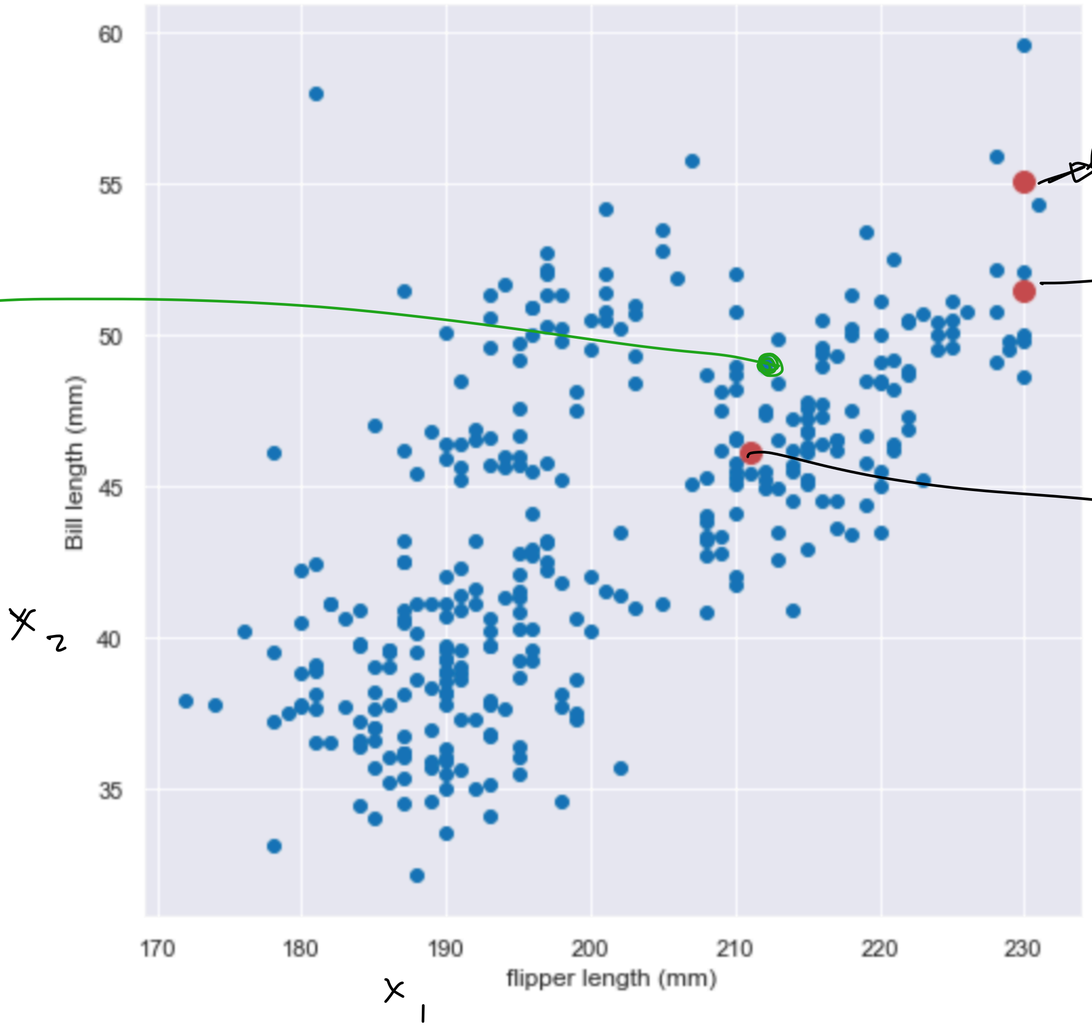
With label



Without label

k-means
Example

- Center initialisation



EPFL

k-Means
Example

- First iteration

$t = 1$

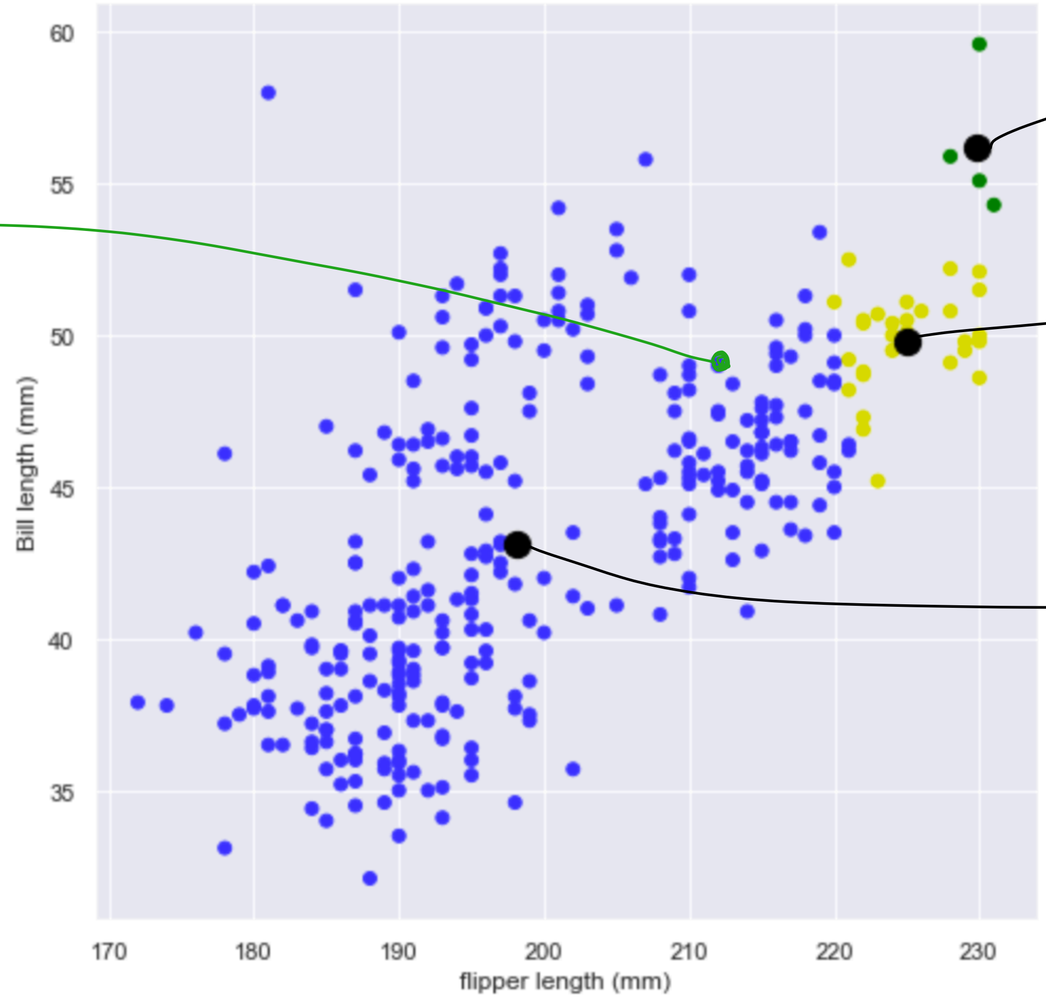
$\min_{j=1,2,3} \| x^i - \mu^j(1) \|_2^2$

$= \| x^i - \mu^2(1) \|_2^2$

$c(i) = 2$

$\downarrow$

x^i belongs cluster 2



$\mu^1(1)$

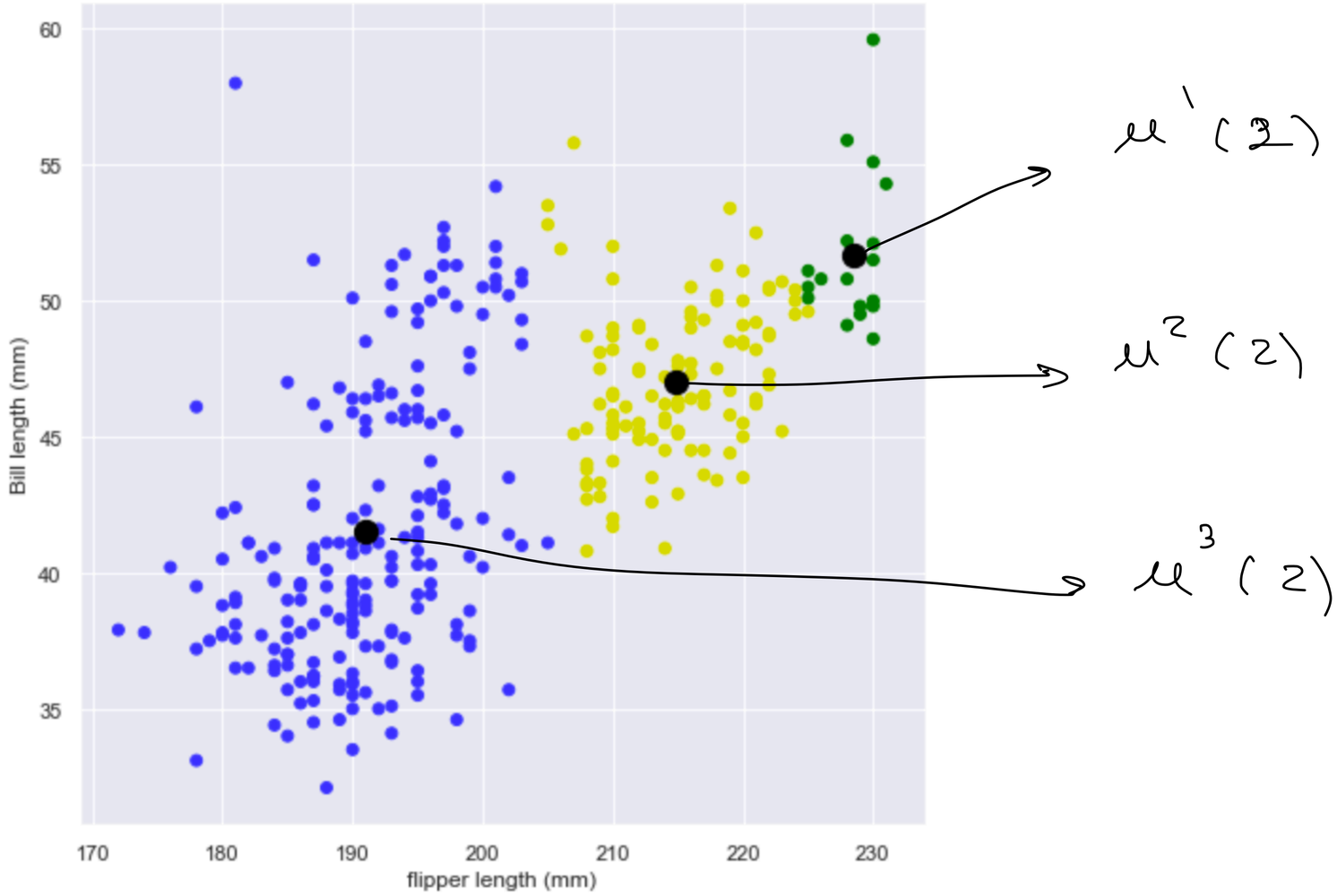
$\mu^2(1)$

$\mu^3(1)$

blue : all points belongs to cluster represented by $\mu^3(0)$

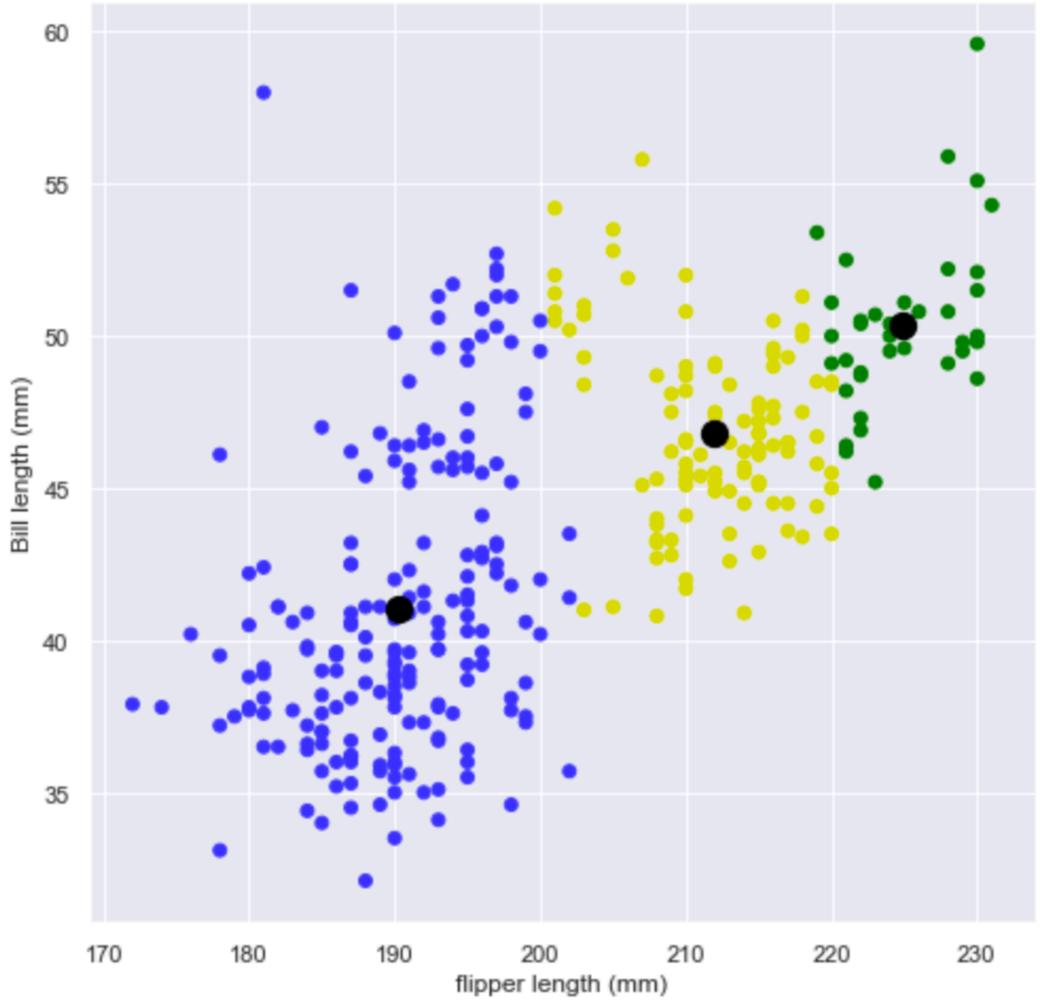
k-Means
Example

- 2nd iteration



k-Means
Example

- 3rd iteration

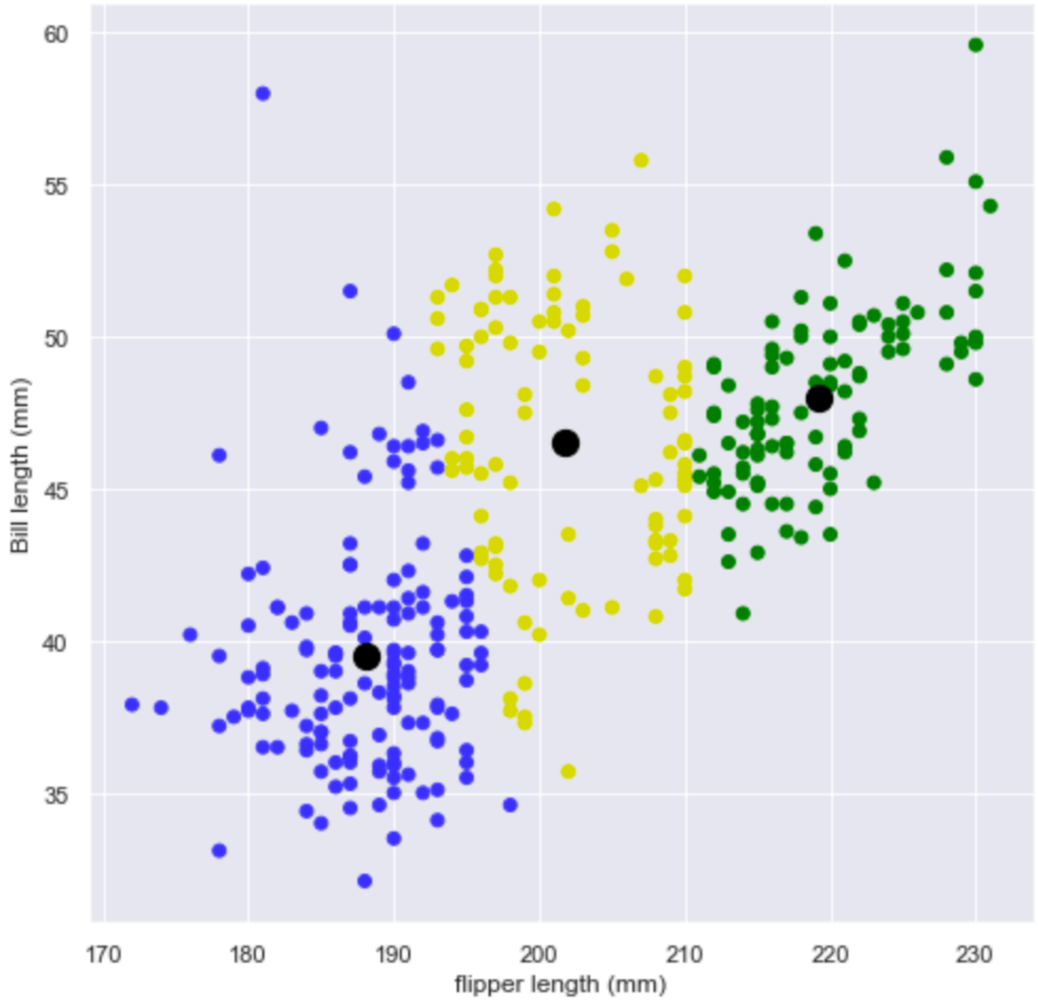


k-Means
Example

- Last iteration

$$\mu^j(t) \approx \mu^j(t-1) \quad \text{for} \quad j = 1, 2, \dots K$$

note: x-axis & y-axis scale are different



Summary - Clustering

Used for understanding data

Examples:
Topic discovery in a large set of documents
Recommendation engines
Guessing missing entries

k-means: an approach to clustering

Easy to implement and to interpret
k-means algorithm usually converges, but possibly to local minima

Brief recap - data statistics

Empirical distribution

$S = \{ \text{blue}(b), \text{yellow}(y), \text{green}(g), \text{red}(r) \}$

empirical distribution $p : S \to \mathbb{R}$

$$\sum_{i=1}^{4} p(s_i) = 1, \quad p(s_i) \geq 0$$

$$p(b) = \frac{3}{12} = \frac{1}{4}, \quad p(y) = \frac{2}{12}, \quad p(g) = \frac{3}{12}, \quad p(r) = \frac{4}{12}$$

EPFL

Brief recap - data statistics

Variance, covariance, correlation

$\{x^i\}_{i=1}^N, \quad x^i \in \mathbb{R}^d$

$\mu_j = \frac{1}{N} \sum_{i=1}^{N} x_j^i$: empirical mean of feature j

$Cov(x_j, x_{j'}) = \frac{1}{N-1} \sum_{i=1}^{N} (x_j^i - \mu_j)(x_{j'}^i - \mu_{j'})$: empirical covariance of feature j, j'

$cov(x_j, x_j) = \sigma_j^2$: empirical variance of feature j.

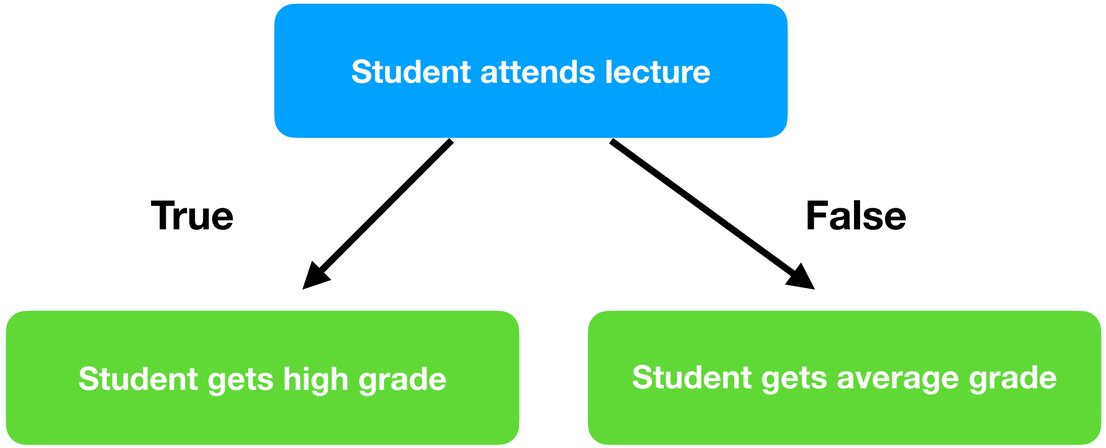
$corr(x_j, x_{j'}) = \frac{cov(x_j, x_{j'})}{\sigma_j \sigma_{j'}}$: empirical correlation between feature j & j'

Decision Trees
Introduction

: an approach to supervised learning

$\{x^i, y^i\}_{i=1}^N$

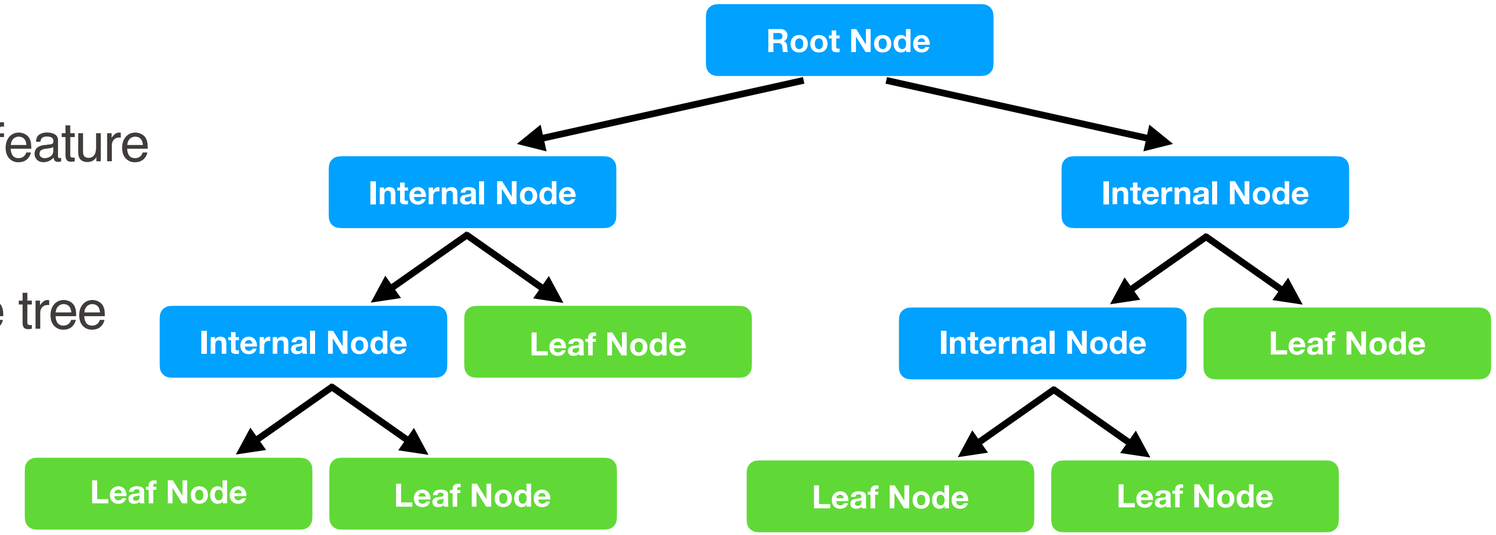
What's a decision tree?



Split data based on the answer to a series of Yes/No questions

Decision Trees
Terminology

- Nodes are checked on a single feature
- Branches are feature values
- Leaves indicate prediction of the tree



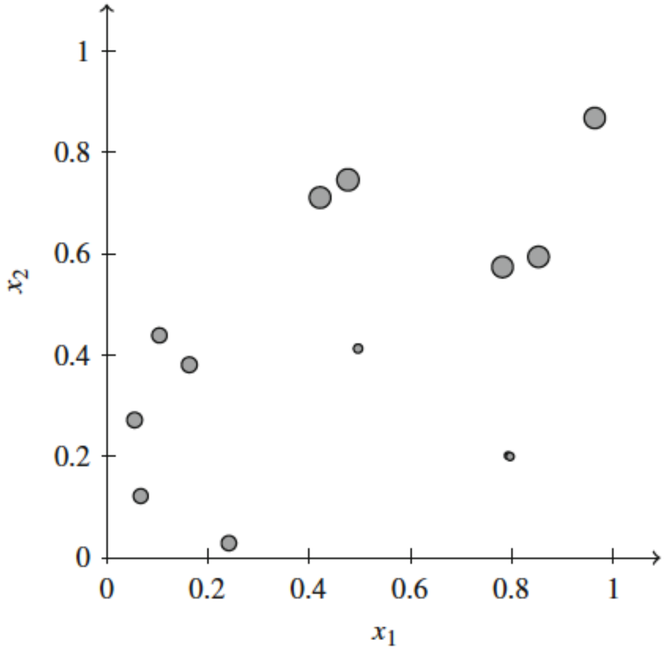
Regression tree

- Example: how to predict the label of an unseen point based on the data?

label

features

y	1.01	0.97	0.97	1.03	0.92	1.97	1.91	2.09	1.93	1.94	0.22	0.26	0.35
x_1	0.06	0.10	0.24	0.16	0.07	0.42	0.96	0.48	0.85	0.78	0.79	0.80	0.50
x_2	0.27	0.44	0.03	0.38	0.12	0.71	0.87	0.75	0.59	0.57	0.20	0.20	0.41



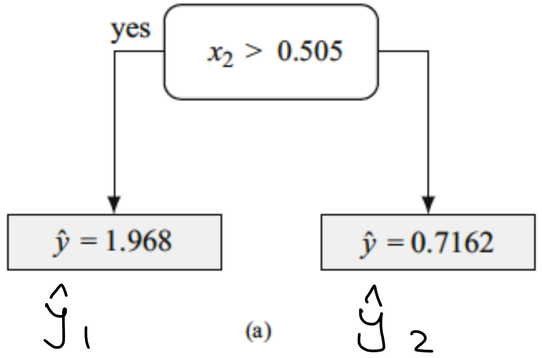
$\hat{y}_1$: average values of $\{y^i\}_{i \in ind_1}$,

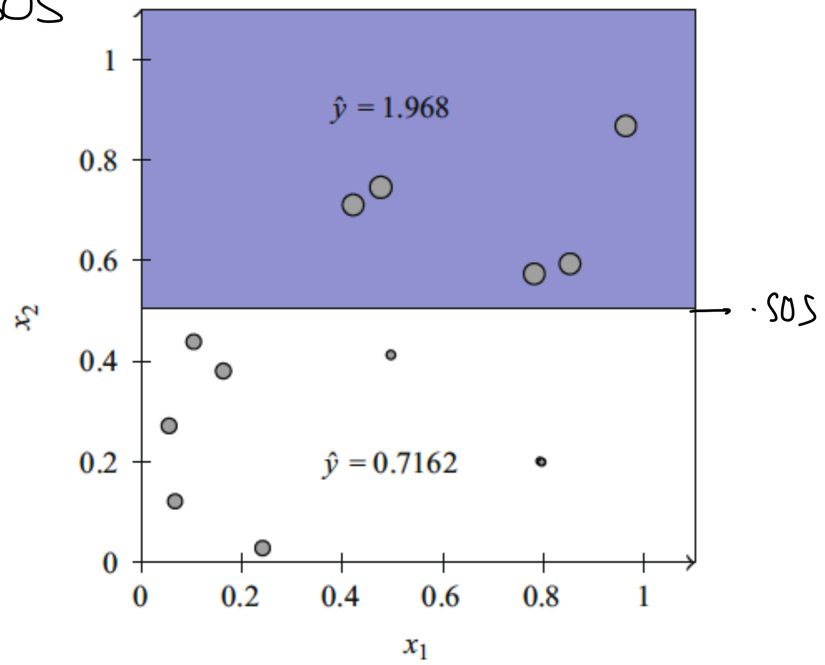
where ind_1 is indices of points whose $x_2^i > 0.505$

$\hat{y}_2$: " " " $\{y^i\}_{i \in ind_2}$,

where ind_2 " " " "

whose $x_2^i \leq 0.505$.





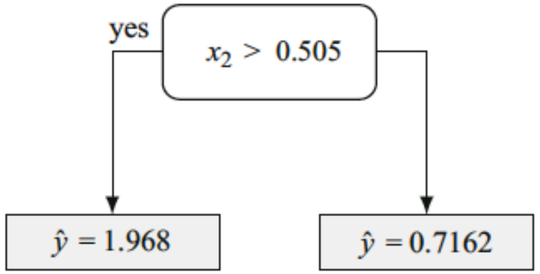
Loss function

- Suppose this is our prediction model. What would be the mean-square loss?

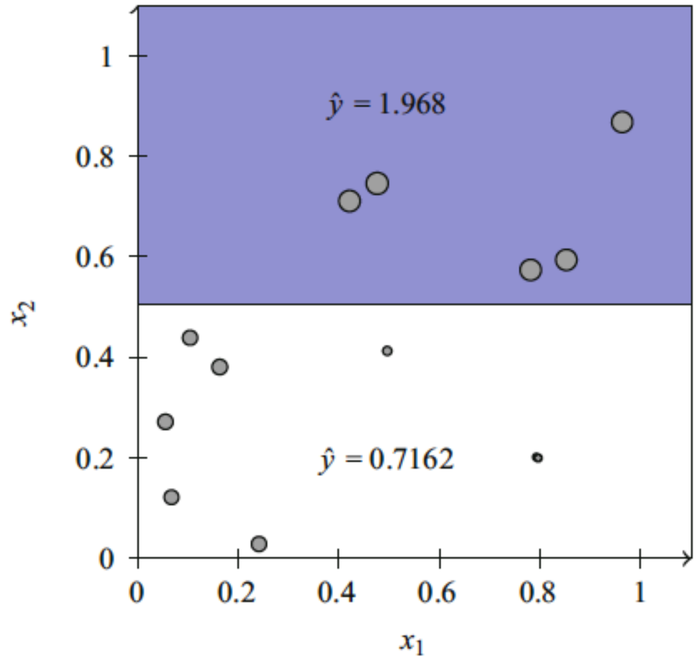
Let ind1 and ind2 denote the set of indices of data points in the top and the bottom regions, respectively

Loss : mean-square error

$$\frac{1}{N}\left[\sum_{i \in ind_1} (y_i - 1.968)^2 + \sum_{i \in ind_2} (y_i - 0.7162)^2\right]$$



(a)



Regression tree example

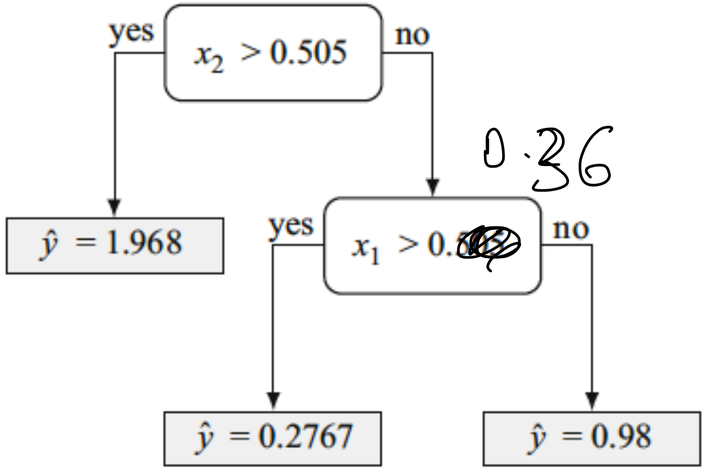
Let ind2 ind3 denote the set of indices of data points in the left and right region, and ind1, as before, denote the set of indices of data points in the top region

Loss: $\frac{1}{N}\Big[$

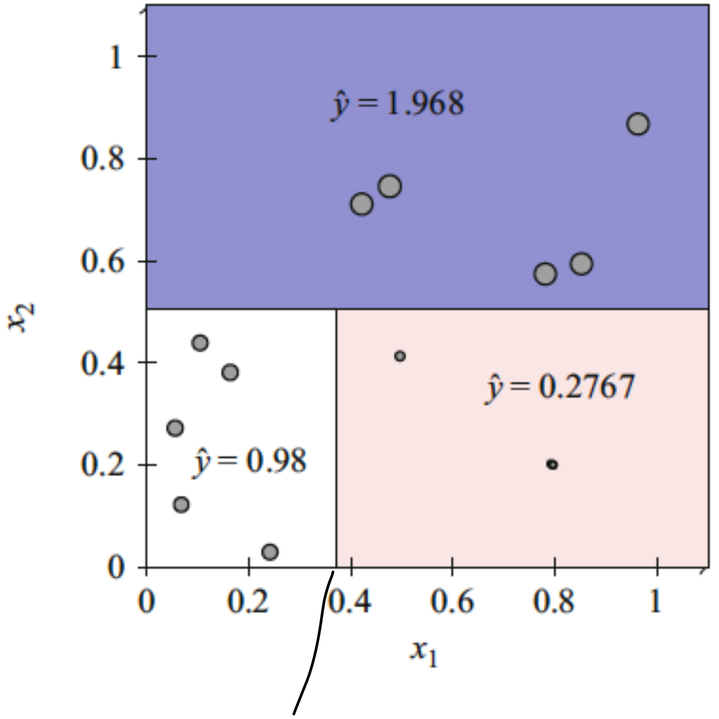
$\sum_{i \in ind_1} (y^i - 1.968)^2 +$

$\sum_{i \in ind_2} (y^i - 0.98)^2 +$

$\sum_{i \in ind_3} (y^i - 0.2767)^2$



0.36



0.36

Regression tree
Loss function

To construct the tree, we have to decide the depth of the tree. And at each node, we have to decide

- which feature to use for a split
- what value of the feature to use for the split

$$\min_{j,s} \sum_{i \in ind_1} (y^i - \bar{y}_{NO})^2 + \sum_{i \in ind_2} (y^i - \bar{y}_{yes})^2$$

where $ind_1 : \{$ indices of $x^i_j \leq s \}$, $\bar{y}_{NO}$ average of corresponding y values.

$ind_2 : \{$ indices $x^i_j > s \}$

to optimize over j, s .. its non-convex ..

Regression Tree

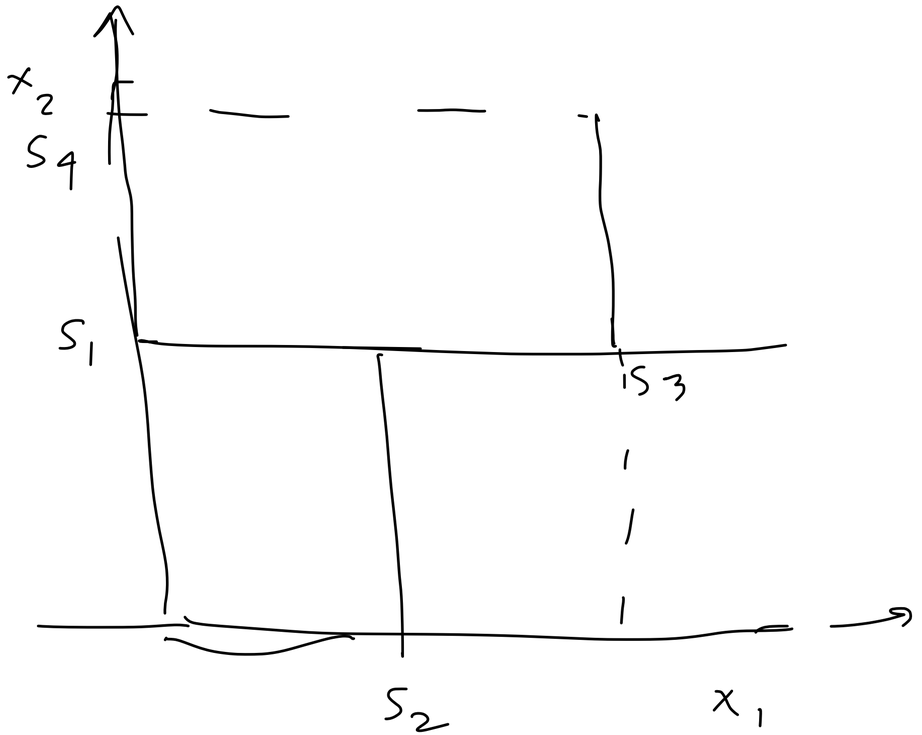
Optimising the loss function - greedy algorithm

The global loss function would tell us which features and what threshold values to use for the entire tree of a given depth. This function is non-convex and hard to optimize.

We apply a greedy approach, where we start adding nodes based on optimizing the loss function at each node.

Note that even the loss function at a given node is non-convex and we can only approximately find a solution

example ... a tree of depth 4 could result in



Regression tree
Stopping criteria

At which depth should we stop growing the tree?

- If the number of data points at leaf nodes become too small, then we often stop
- If the loss at a depth d1 is close to the loss at depth d2 = d1+1, then we often stop
  - As the objective is non-convex and greedy approach gives us an approximate solution, it's possible that the loss would decrease after a few iterations, even though in the next iteration it did not decrease

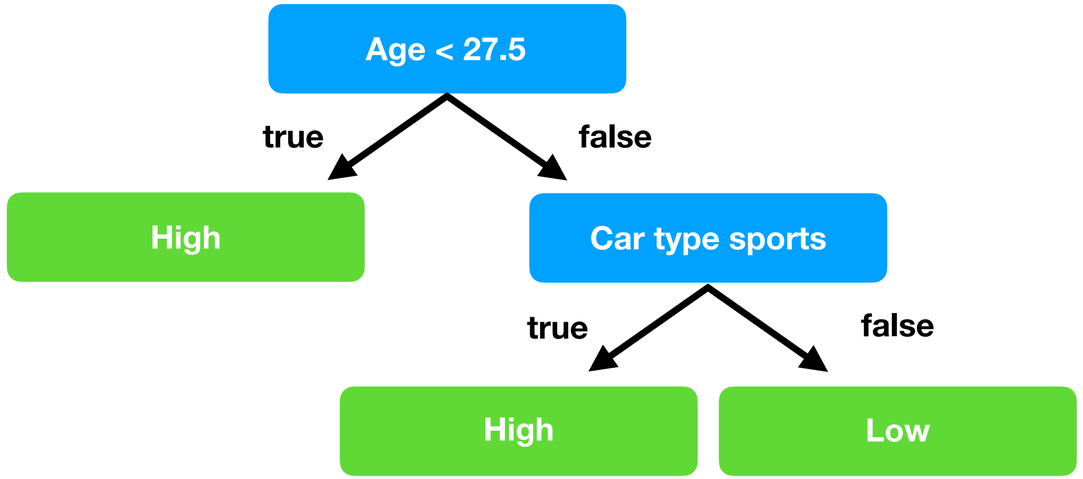
Decision Trees for classification

Called classification trees

$\{x^i, y^i\}_{i=1}^{N}$ $\quad y^i \in \{high, low\}$ risk

$N = 6$, x^i_1 : age, x^i_2 : car type

Continuous Feature	Categorical Feature	Class label
Age	**Car type**	**Risk**
23	family	high
17	sports	high
43	sports	high
68	family	low
32	family	low
20	family	high



Classification trees

Performance metric and loss function

Given a classification tree, we can evaluate its performance by forming the confusion matrix

We can measure the accuracy, error rate, etc.

However, finding the tree that minimises a given metric is a hard optimisation problem

To address this, we use an alternative measure of performance and use a greedy approach based on this measure

Classification Trees

Example - constructing the tree

Let’s consider a tree of depth 2. We have to address

Which feature to use at each depth to do a split?

For the continuous feature, at what value to do a split?

For the categorical feature, which category to use for the split?

Continuous Feature	Categorical Feature	Class label
Age	**Car type**	**Risk**
23	family	high
17	sports	high
43	sports	high
68	family	low
32	family	low
20	family	high

Classification trees

Greedy approach: choose a feature and the split sequentially based on minimising a performance metric, for example, the **Gini impurity** of a node

Gini impurity of a leaf node: based on empirical probability of class

K classes

$$g = \sum_{l=1}^{K} \sum_{l' \neq l} p_l p_{l'} = \sum_{l=1}^{K} p_l (1 - p_l)$$

$K = 3$, p_l : prob. of class l

$$g = (p_1 p_2 + p_1 p_3) + (p_2 p_1 + p_2 p_3) + (p_3 p_1 + p_3 p_2)$$

$K = 2$

$$g = p_1 p_2 + p_2 p_1 = 2 p_1 p_2$$

Gini impurity of a node

Weighted sum of the gini impurity of the two leaf nodes associated with the node

Note: Criteria other than gini index (such as entropy) are also used for node split

Classification Trees

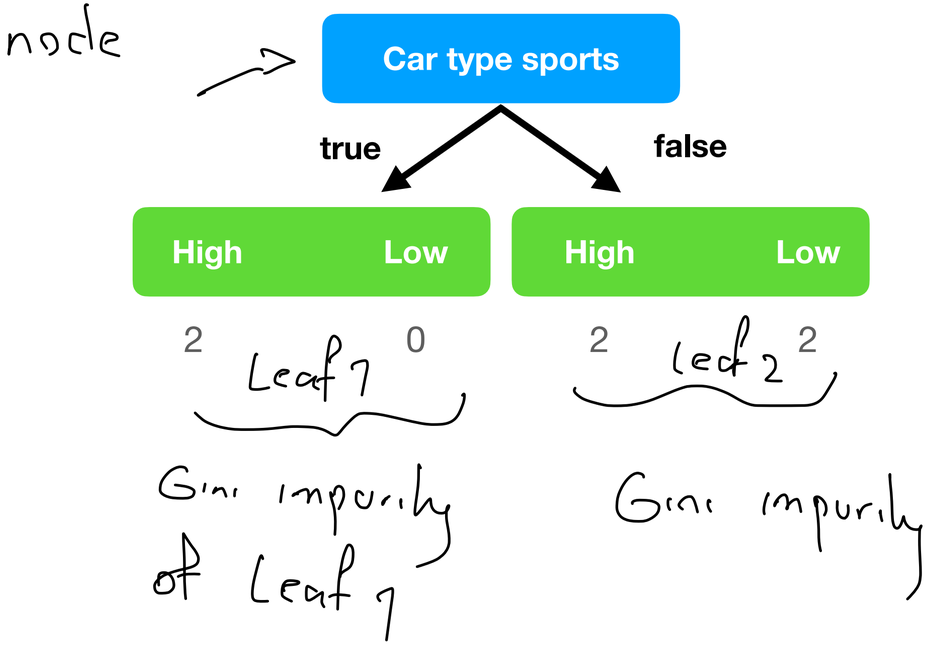
Example - constructing the tree

We will apply Gini impurity to construct the classification tree

Continuous Feature	Categorical Feature	Class label
Age	**Car type**	**Risk**
23	family	high
17	sports	high
43	sports	high
68	family	low
32	family	low
20	family	high

Classification trees

Criteria for choosing a feature and a split



Continuous Feature	Categorical Feature	Class label
Age	**Car type**	**Risk**
23	family	high
17	sports	high
43	sports	high
68	family	low
32	family	low
20	family	high

$$2\,\frac{2}{2}\times\frac{0}{2}=0 \qquad 2\left(\frac{2}{4}\times\frac{2}{4}\right)=\frac{1}{2}$$

Gini impurity of node :

$$\frac{2}{6}\times 0 \;+\; \frac{4}{6}\times\frac{1}{2} \;=\; \frac{1}{3}$$