Unsupervised learning
18.11.2024

Outline

- Hour 1
  - Quiz discussion, review of last lecture
  - Introduction to unsupervised learning:
    - Principal component analysis (PCA)
- Hour 2:
  - PCA continued
  - K-means

Review of data statistics, ex: Quiz 1 grades

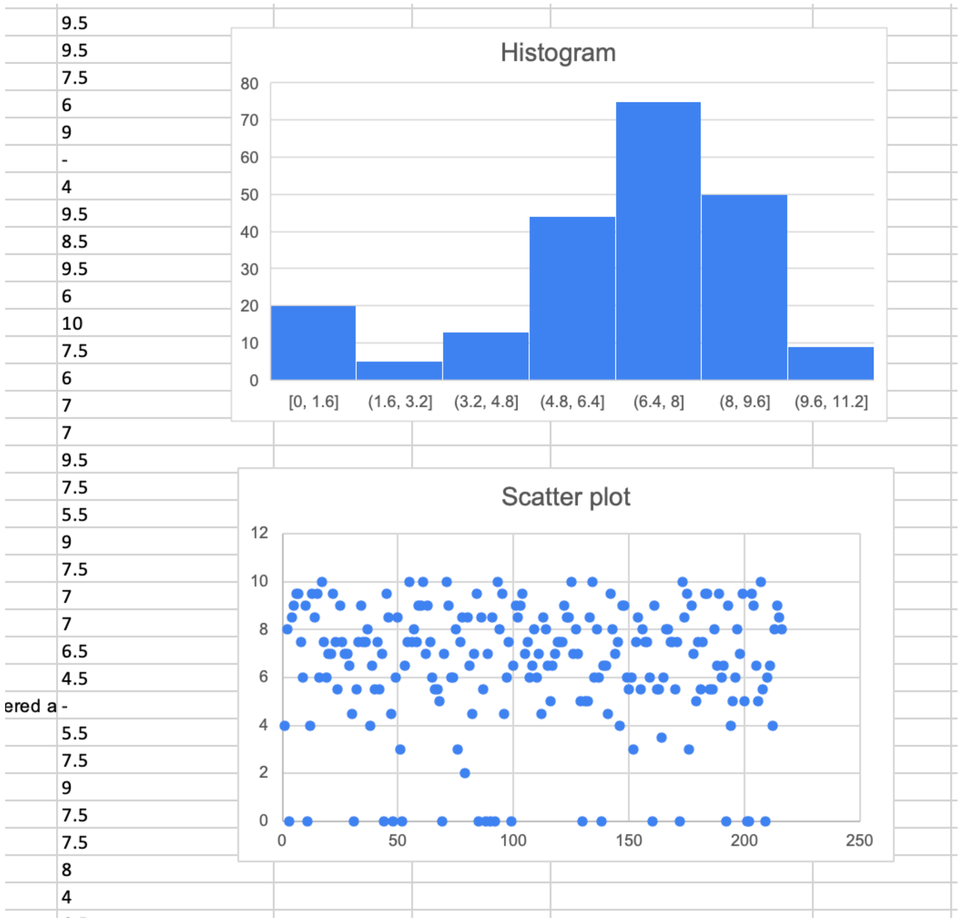
Mean: 7.16, Standard deviation: 1.75, Mode: 7.5

Quartiles:

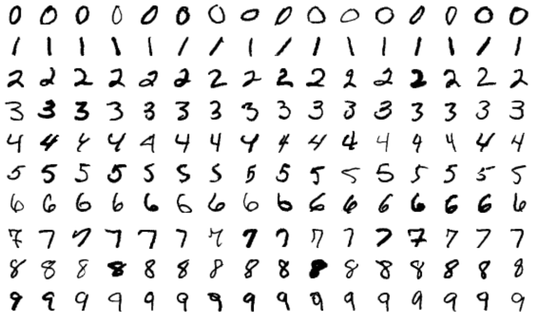
1st: 6, 25% of grades is below this number

2nd (median): 7.5, 50% of grades is below this number

3rd: 8.5, 75% of grades is below this number



Review last time



Consider MNIST data

Each sample is an image of size

Features:

Neural networks (NN)

Consider a single layer NN with 10 nodes in the first layer. How many parameters need to be determined to specify this NN?

Convolutional neural networks (CNN)

Consider a single layer CNN with 5 filters, each of which is 3x3.

How many parameters need to be determined to specify this CNN?

Example applications of NN in IGM

Prof. Colin Jones: Predictive Control Lab

Goal: develop a dynamical model of the temperature evolution in a building for control (Control objective: minimize energy consumption while ensuring comfort of occupants)

Introduction

Issues of classical models

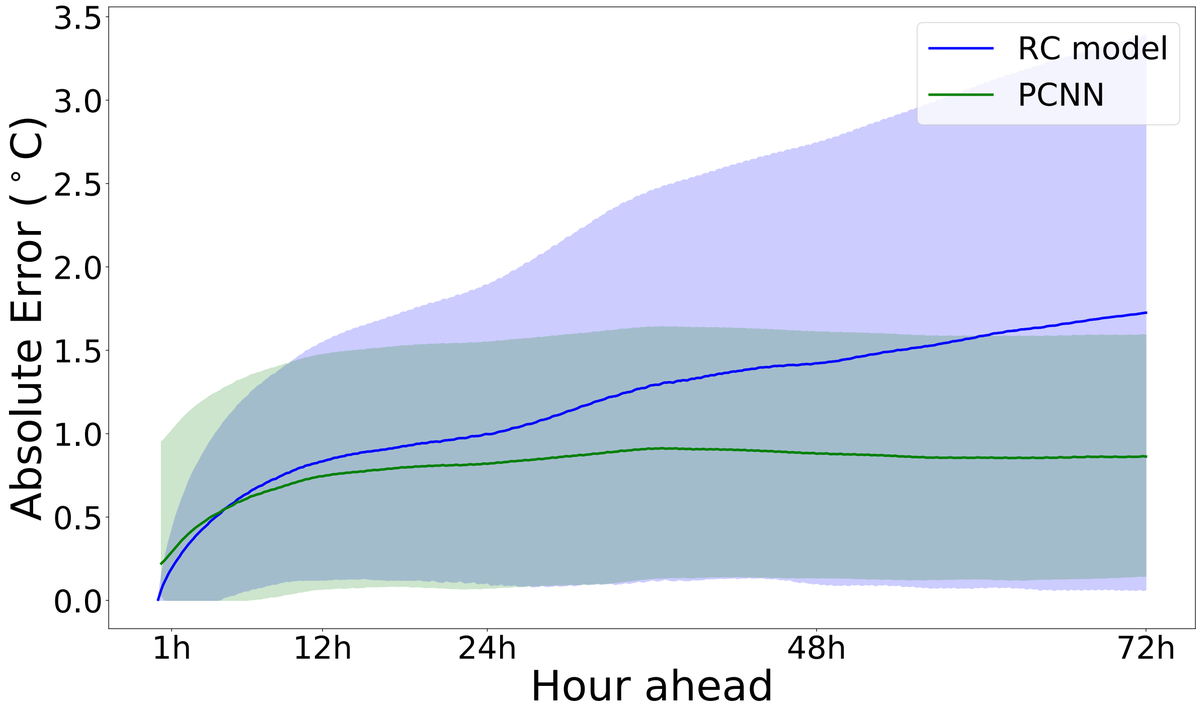
- Classically, people use **physics-based models**, built from first principles
  - Follow the **laws of physics**
  - Large **engineering overhead**

… but each building is different!

… can we leverage data in black-box methods?

© Steiner SA, Geneva

© Zooey Braun, Stuttgart



Introduction
Unsupervised learning

Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a **data set with no pre-existing labels** and with a minimum of human supervision.

in the next techniques, we don’t use labels anymore
Note: the objective is vague but we will consider 2 concrete instances

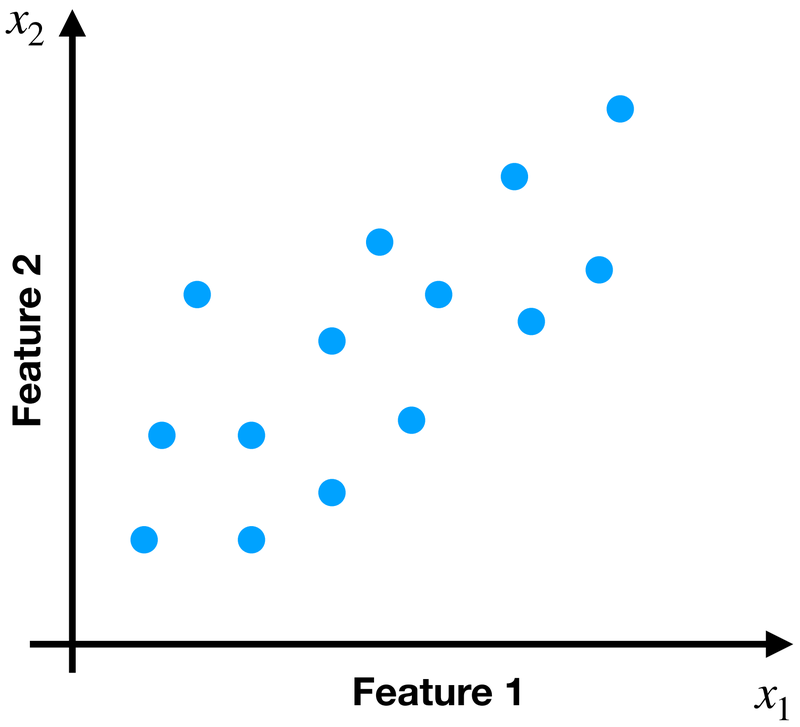
Dimensionality reduction

Through

principal component analysis

Motivation
Intuition

We have samples described by a series of features



We want to find a smaller set of new features that explain our sample because:

Less features is easier to visualize

Some of the current feature can be redundant

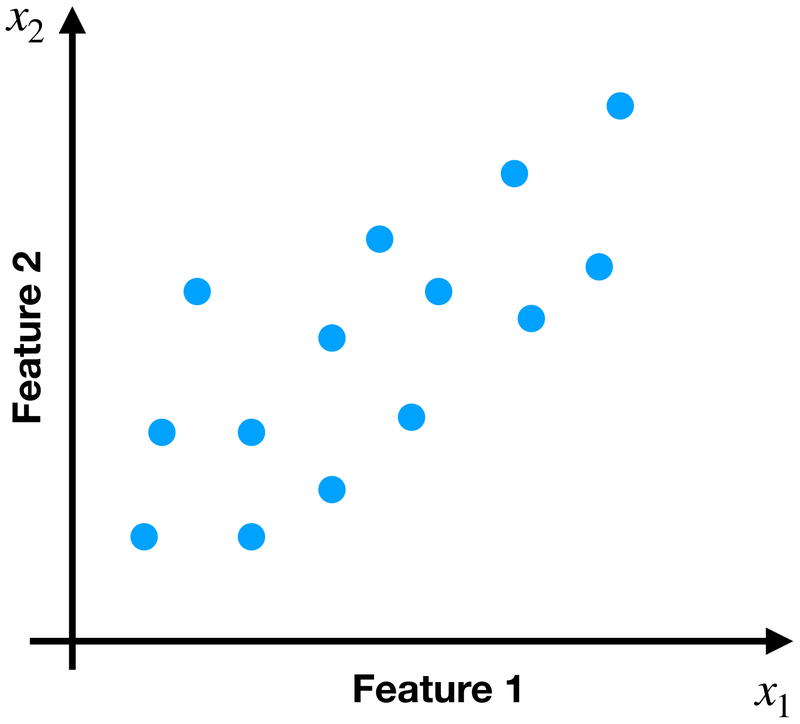
Some of the current features are not very useful to describe our samples

Principal component analysis (PCA)

Approach to dimensionality reduction

How to find this smaller set of new features ?

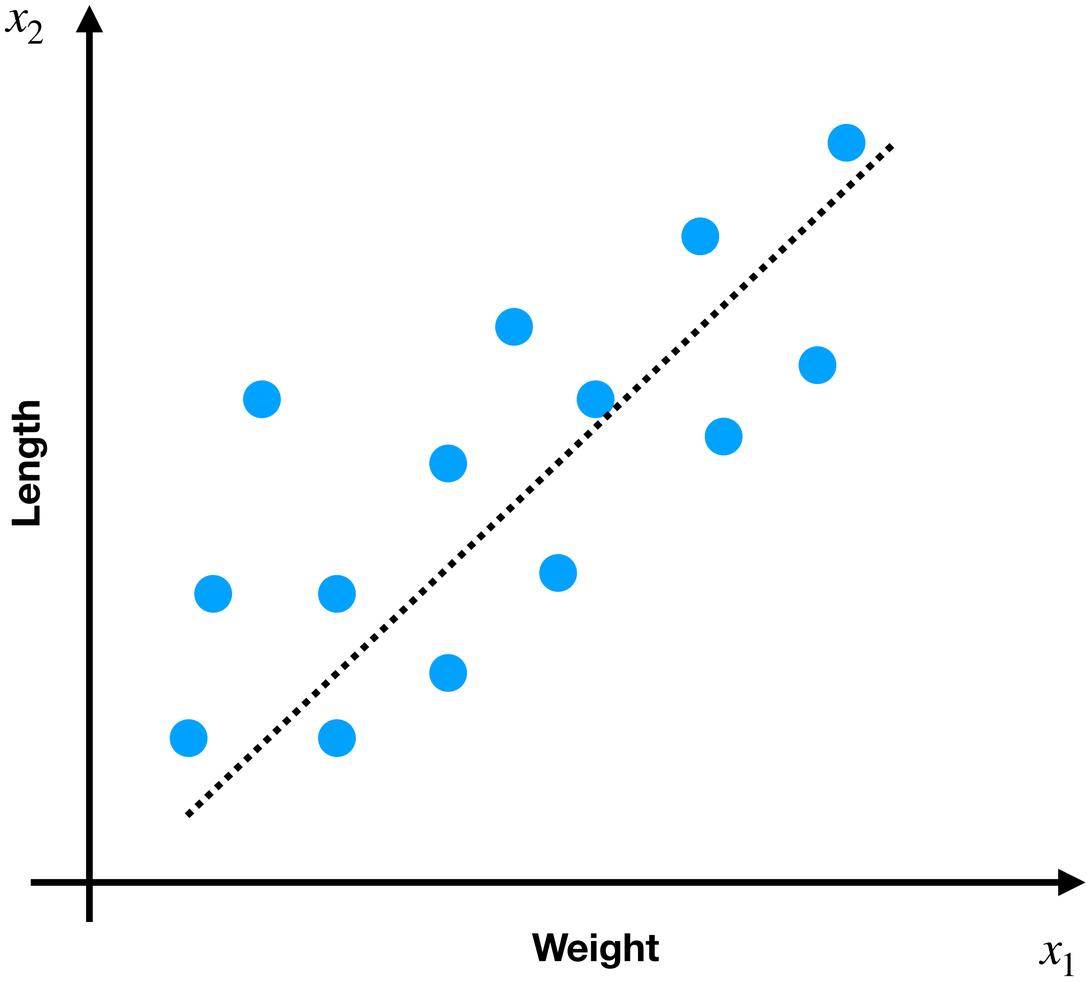
PCA : Find the best **linear combination** of features to create new features that explain our samples better



PCA

Projection of points onto a lower dimensional subspace

How to find this smaller set of new features ?



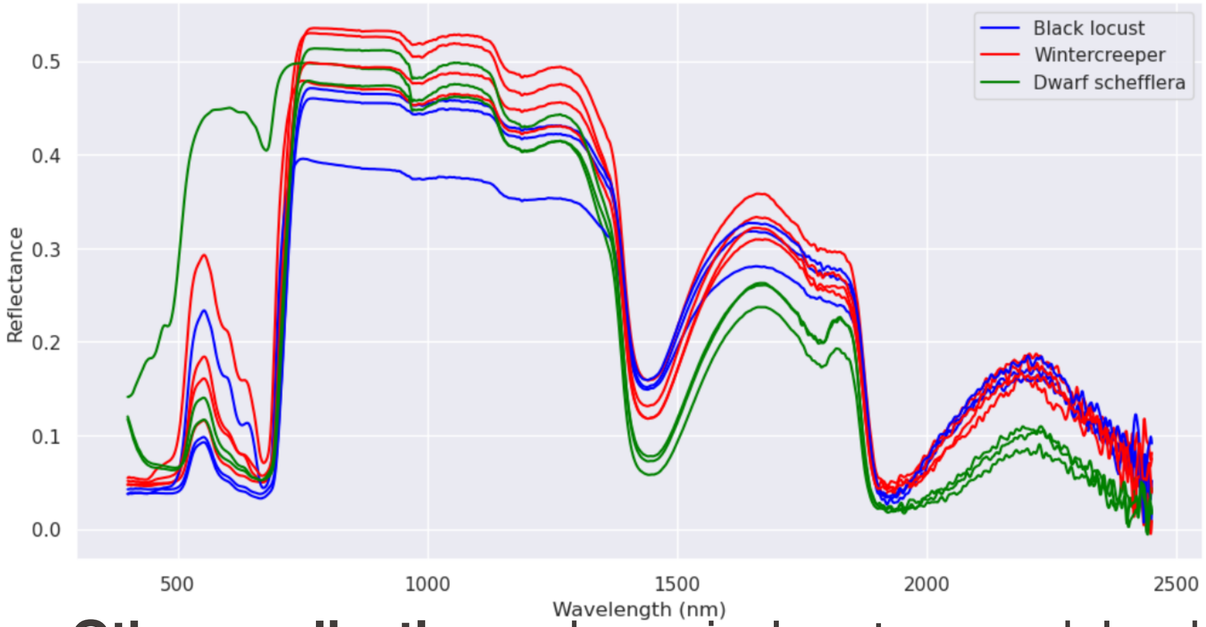
$$w_1 x_1 + w_2 x_2$$

The new feature

A mix of length and weight that describe our samples better

PCA - example application in python

- **Dataset:** Leaves' optical reflectance measured at each nm in the range of 450-2500 nm Wavelength, see more details about the experiment here
- **Question:** what is the size of feature vector for each leaf?
- **Goal:** can we reduce the dimensionality of feature vector and still capture the distinguishing features of each leaf



- **Other applications:** dynamical system model order reduction, audio compression for recommendation algorithms, text processing for news recommendation

Finding distance of a point to a subspace

Subspace

Distance of a point to a subspace

Projection of a point onto a subspace

Distance of a point onto a subspace

Choosing orthonormal vectors

Projection of a point onto a subspace

Find a subspace to minimize average distance of data to it

Sum of distances of all data points to a subspace

PCA chooses a subspace to minimize

Formulation of PCA objective using the data matrix

Standardize data:

Frobenius norm of a matrix:

PCA objective:

Eigenvalue decomposition of a symmetric matrix

Solution to PCA using eigenvalue decomposition

Principal components

Example: projection of data onto 2 principal components

PCA
Pseudo-code

1. **Standardize data**

2. **Eigenvalue decomposition of the data covariance matrix**

3. **To reduce the dimension to**

4. **Compressed feature matrix**

PCA example - distinguishing texts

Defining features

Each data sample is a document

There are d unique words in all the documents

Feature is positive for document if word is in document

TFIDF feature definition for documents

Term frequency of word … in document …

Document frequency of word ….

TFIDF for each word ….

PCA example

Based on “Principal Component Analysis” lecture of Stanford EE104: https://ee104.stanford.edu/lectures/pca.pdf

Distinguishing text: *The Critique of Pure Reason* by Immanuel Kant and *The Problems of Philosophy* by Bertrand Russell

Dimensionality reduction
Other techniques

There are several other techniques for dimensionality reduction :

Linear discriminant analysis (LDA)

Generalized discriminant analysis (GDA)

T-distributed Stochastic Neighbor Embedding (t-SNE)

Autoencoders

Autoencoder
Introduction

- An autoencoder is a type of neural network often used for dimensionality reduction.
- Autoencoders are trained in an **unsupervised** manner, by minimizing the reconstruction error / loss $\sum_{i=1}^{N} L\left(x^i, \hat{x}^i\right)$
- Example: Squared error

Autoencoder

Autoencoder vs. PCA

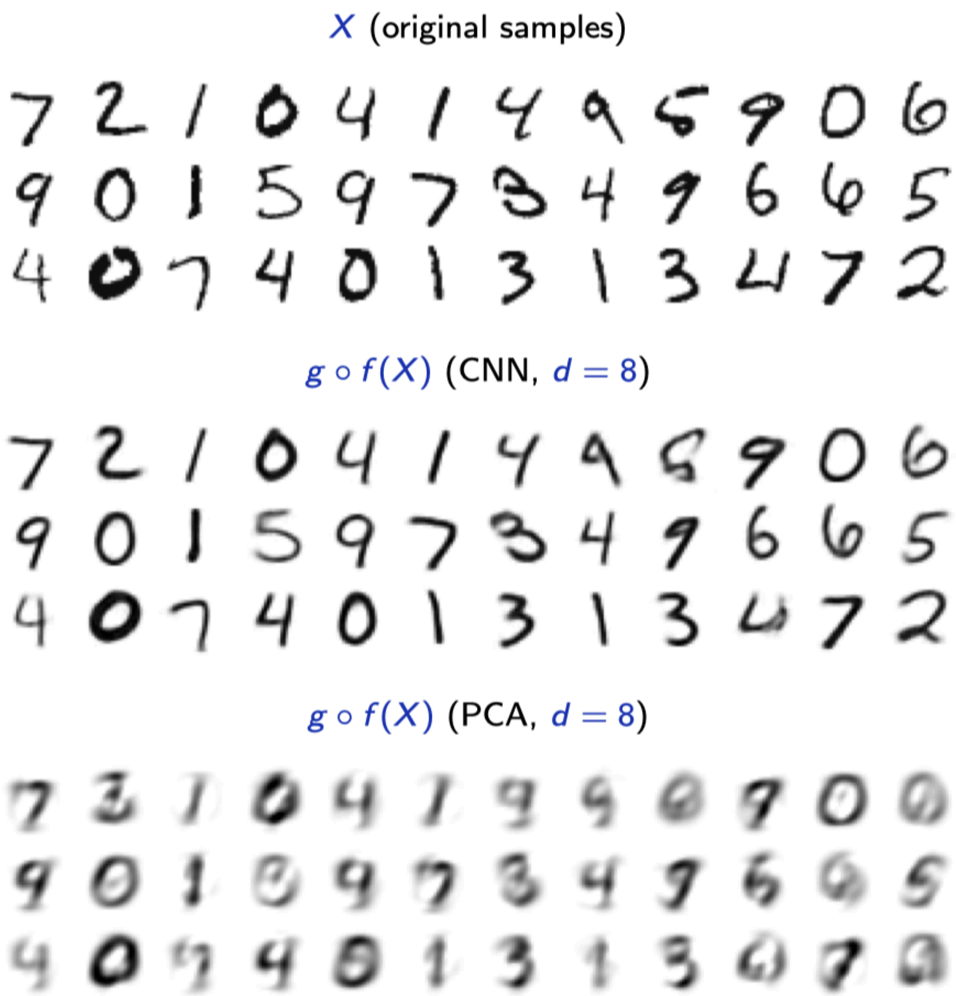


Image credit: F. Fleuret, Deep Learning (EPFL)

Top: Some examples of the original MNIST test samples

Middle: Reconstructed output from an auto-encoder with a latent space of 8 dimensions

This auto-encoder uses convolutional layers, and was trained on the MNIST training set

Bottom: Reconstructed output from PCA with 8 reduced dimensions

Summary - dimensionality reduction

Used for

- Exploratory data analysis
- Visualizing data
- Help reduce overfitting by reducing feature dimension

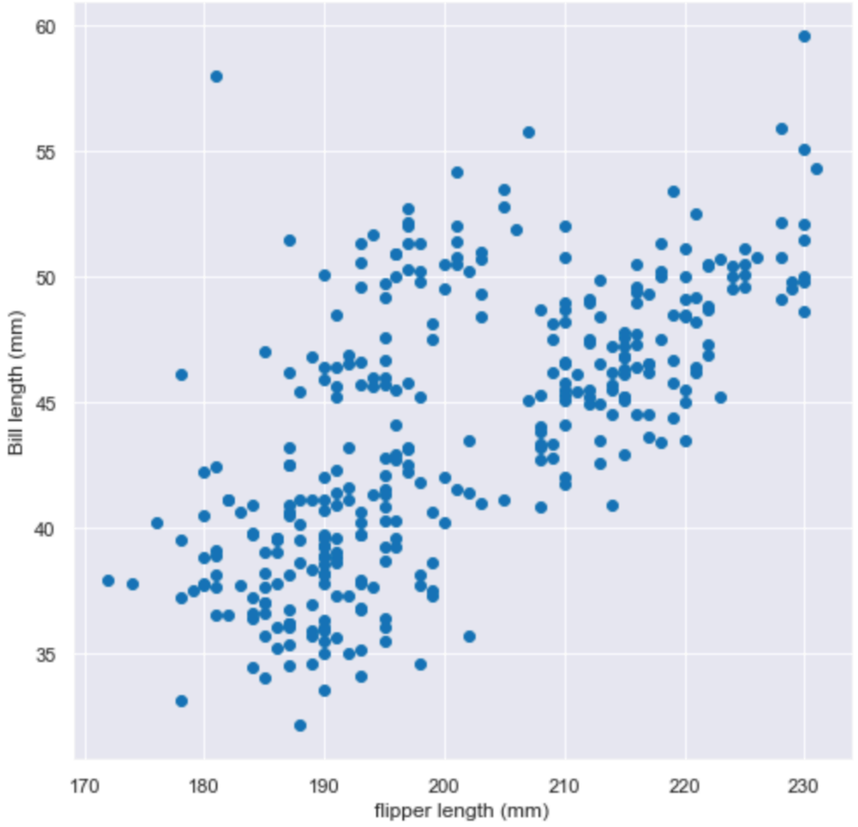
PCA: an approach to dimensionality reduction

- Projects data onto a linear subspace
- Useful in case there is approximately linear dependence between different features
- Easy to compute
- Connection to singular value decomposition (see Problem set 2)

Clustering

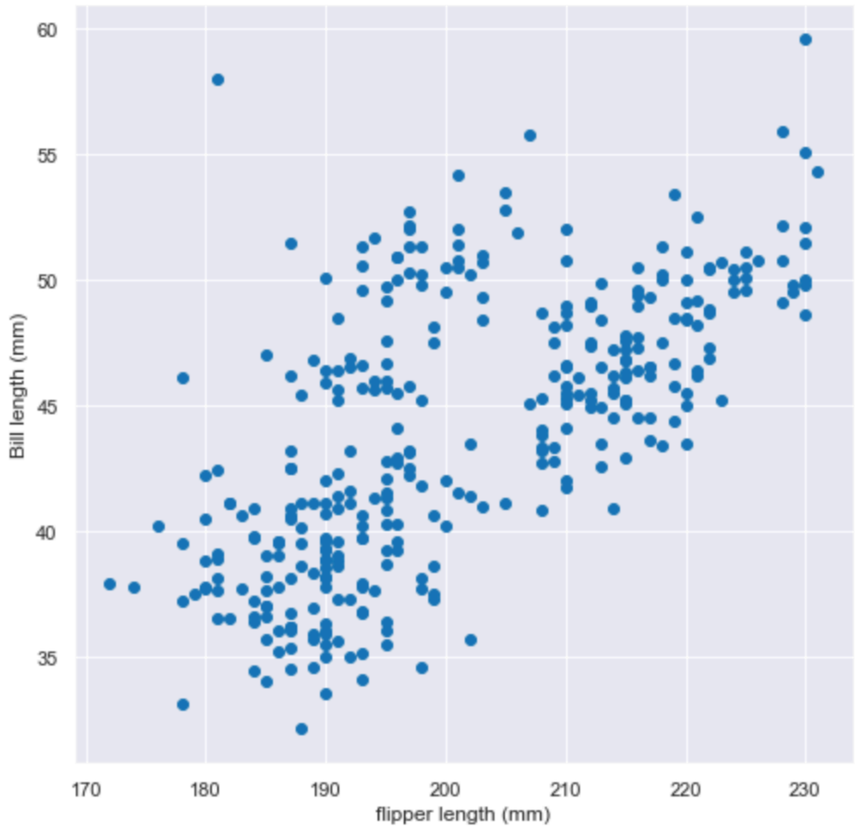
K-means
Example

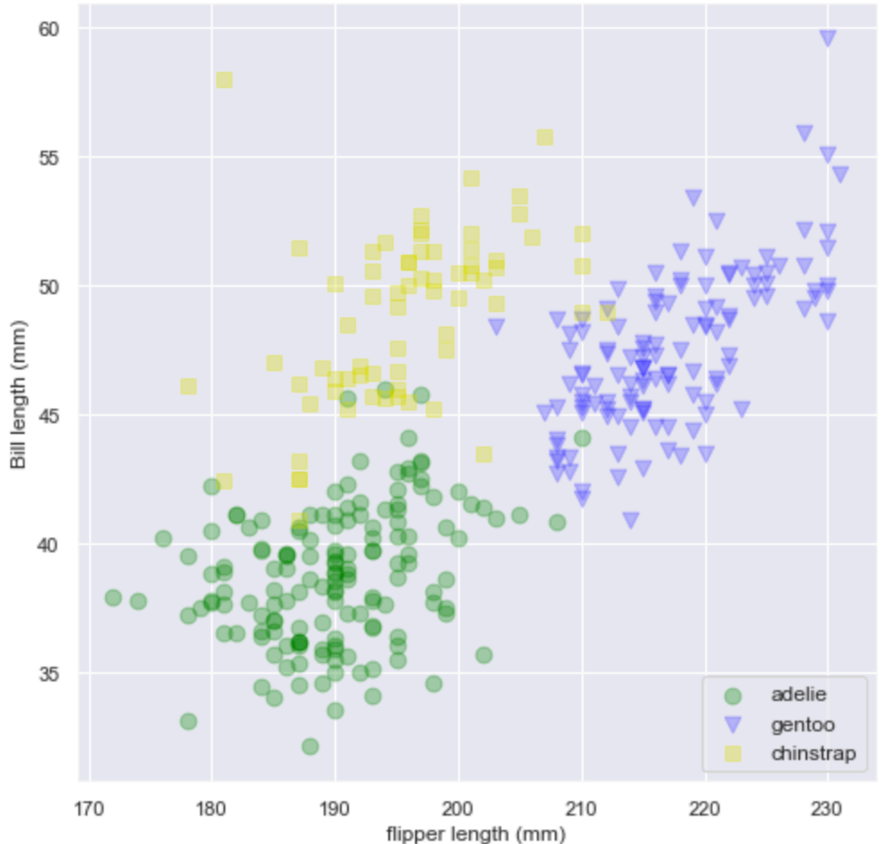
- Palmer Penguins dataset
  - Features: Flipper length and Bill length
  - Can we identify k species based on this data?



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 $\mu_1, \mu_2, \ldots, \mu_k$ representing the center of each cluster and add the other data points to the nearest cluster center.

k-means
Preliminaries

- A single representative point for data:

- Example: suppose we want to have one representative point
  - we use a mean-squared
  - we use an absolute value loss

k-Means
Approach

- Choose k clusters to represent data
- Determining the cluster a single point … belongs to
- Determining the cluster centres to minimize the distance of each point to its assigned cluster

k-Means heuristic algorithm

Algorithm -

1. Initialize $\{\mu_1, \mu_2, \ldots, \mu_k\}$ (e.g., randomly)
2. While not converged
    1. Assign each point …. to the nearest center
    2. Update each center μ_j based on the points assigned to it

- **Step 2.1:** For each point …, compute the Euclidean distance to every center $\{\mu_1, \mu_2, \ldots, \mu_k\}$
  - Find the smallest distance
  - 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

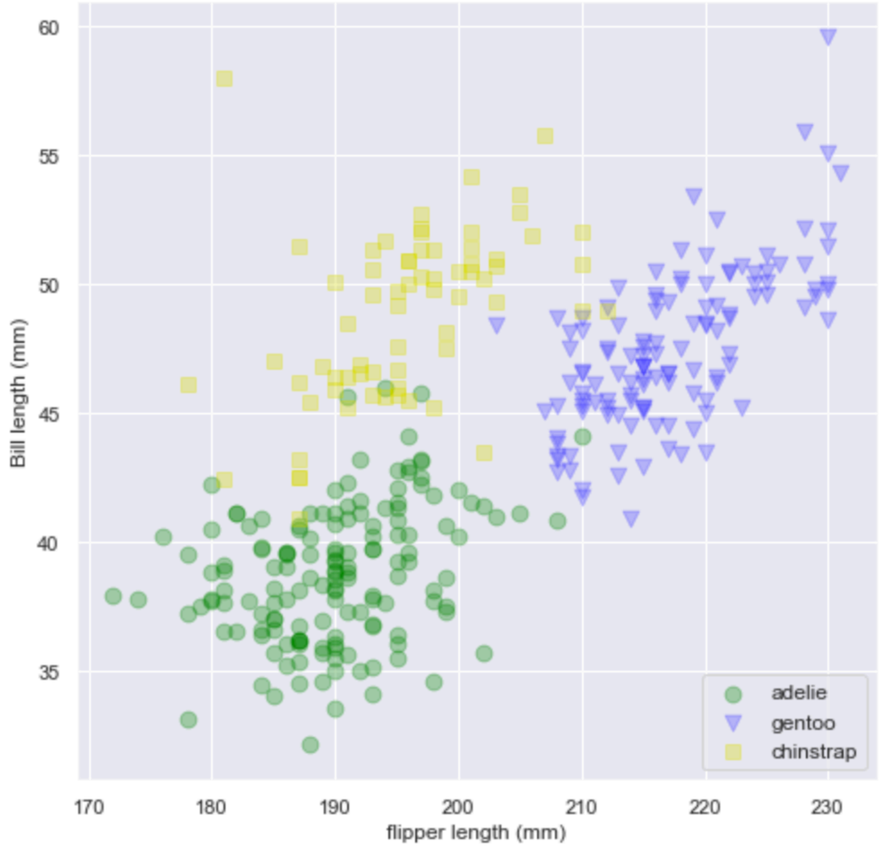
k-means
Algorithm - Convergence

- **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

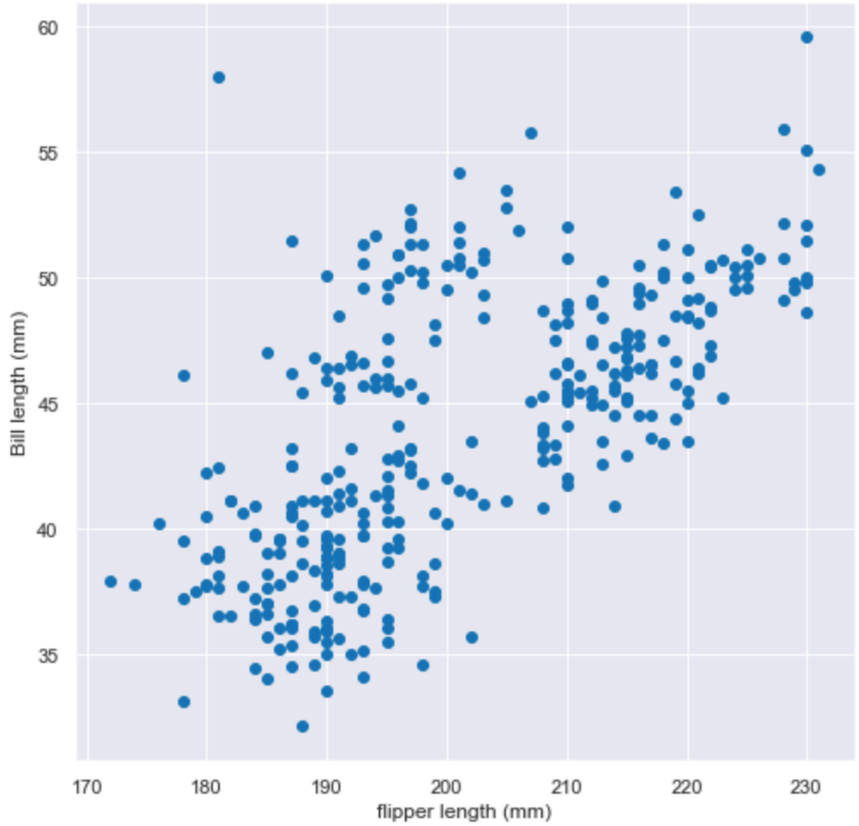
- **K-Means does not** always converge to the **best solution**
  - Non-convex optimization problem

k-means
Example

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



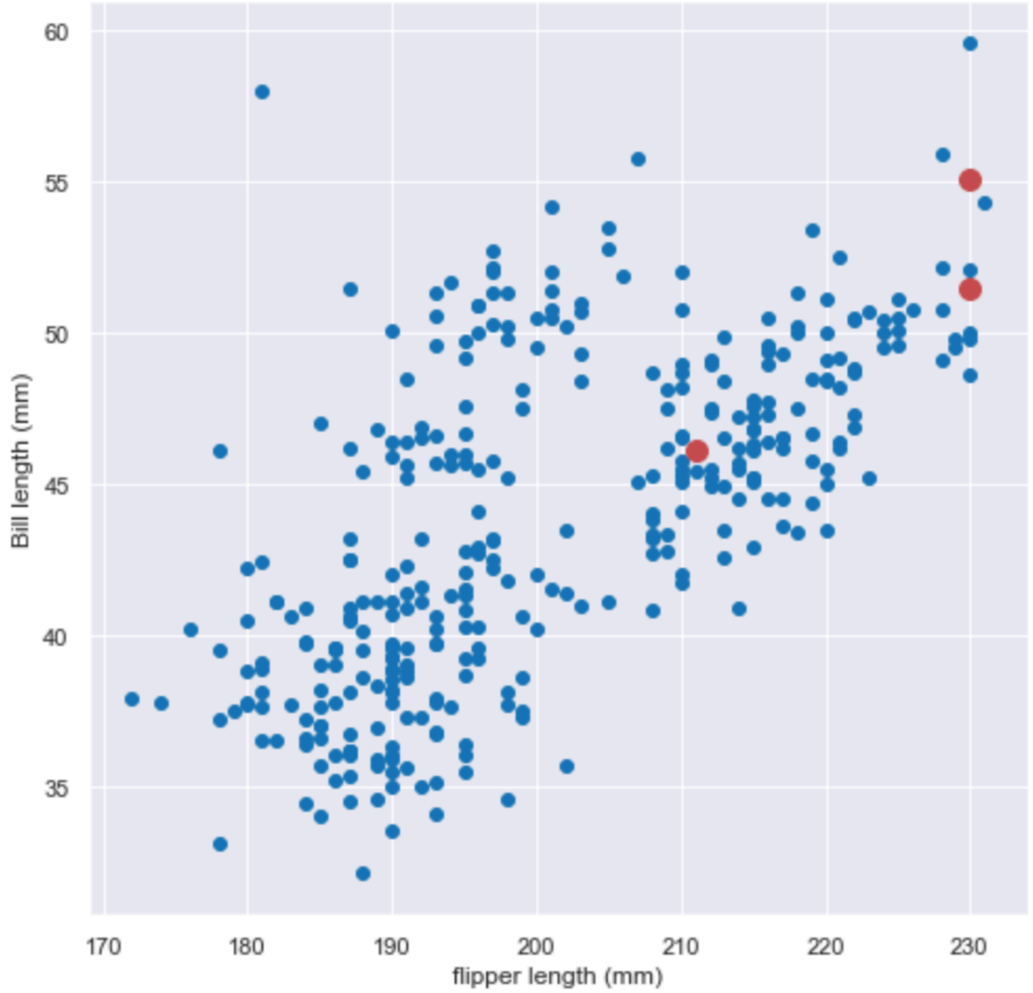
With label



Without label

k-means
Example

- Centroid initialisation

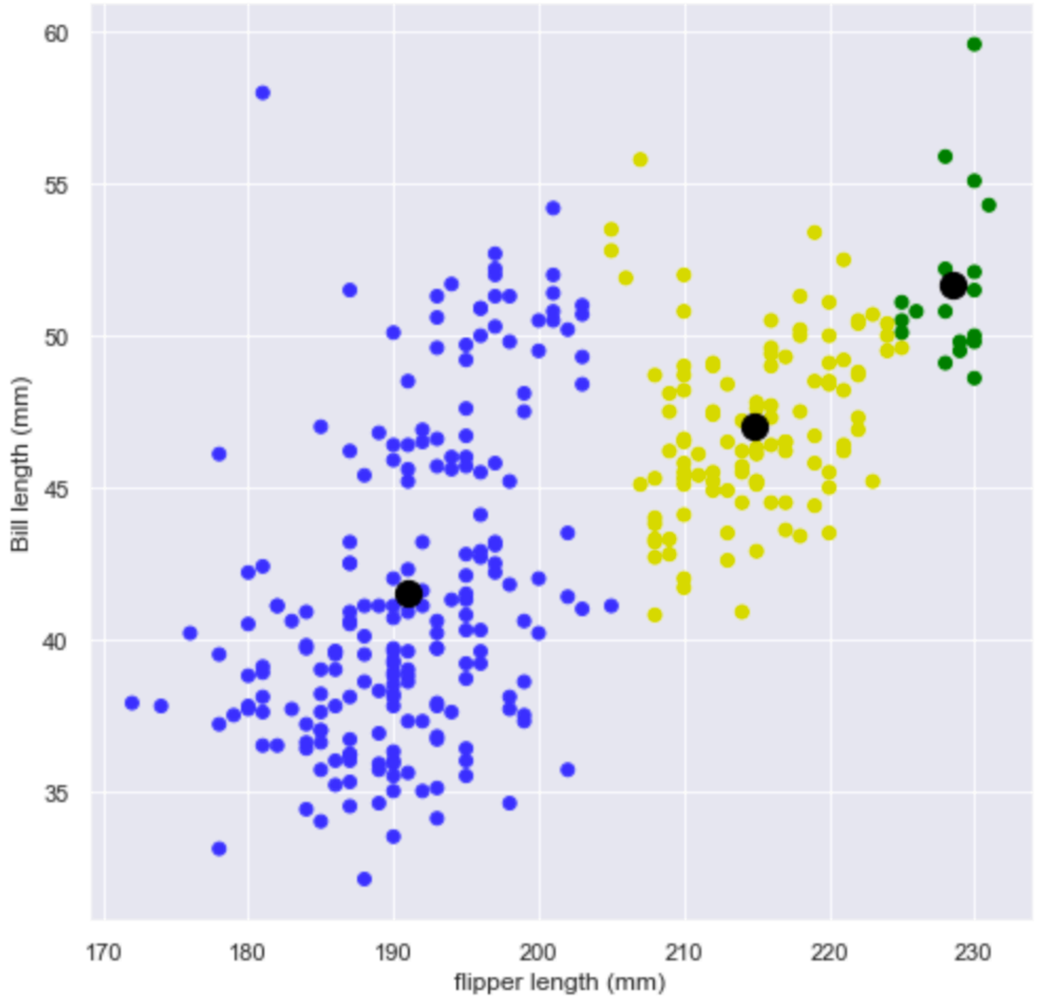


k-Means
Example

- First assignment

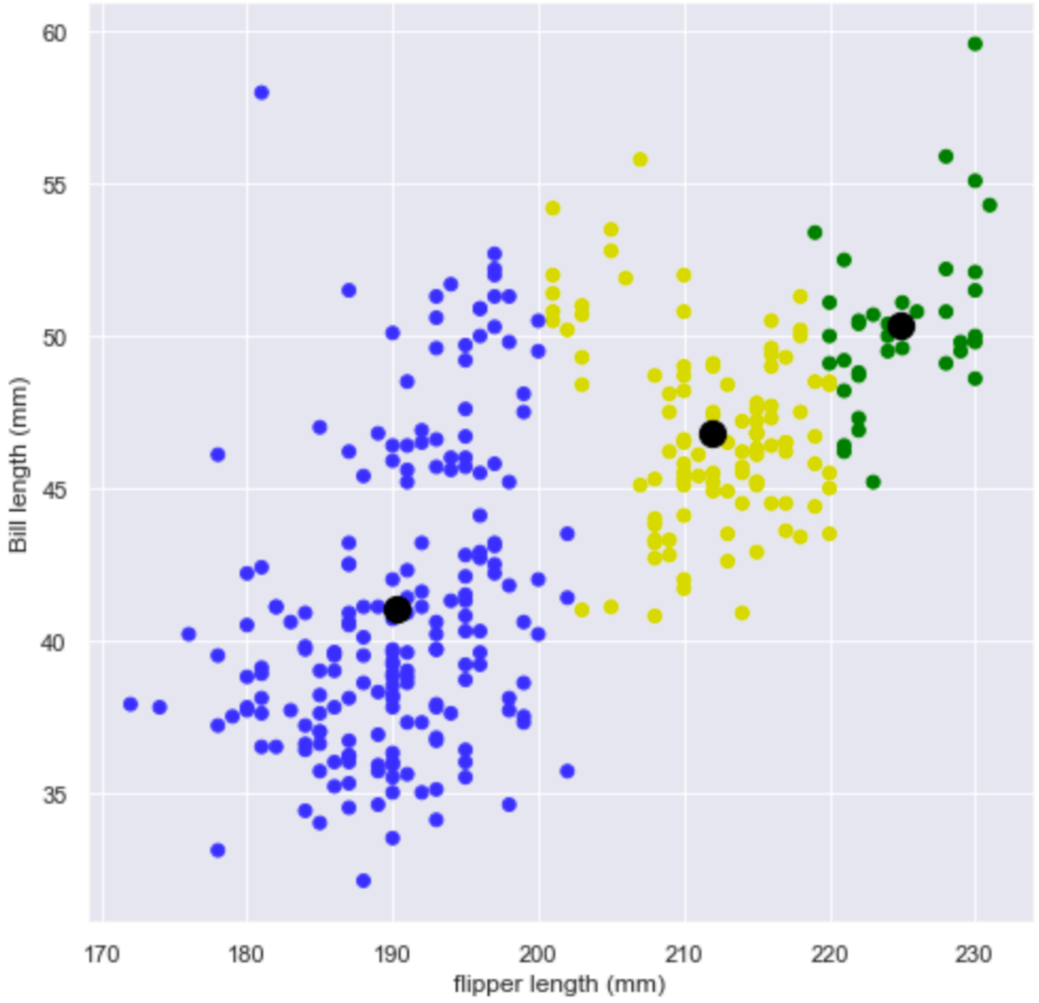
k-Means
Example

- Next assignment



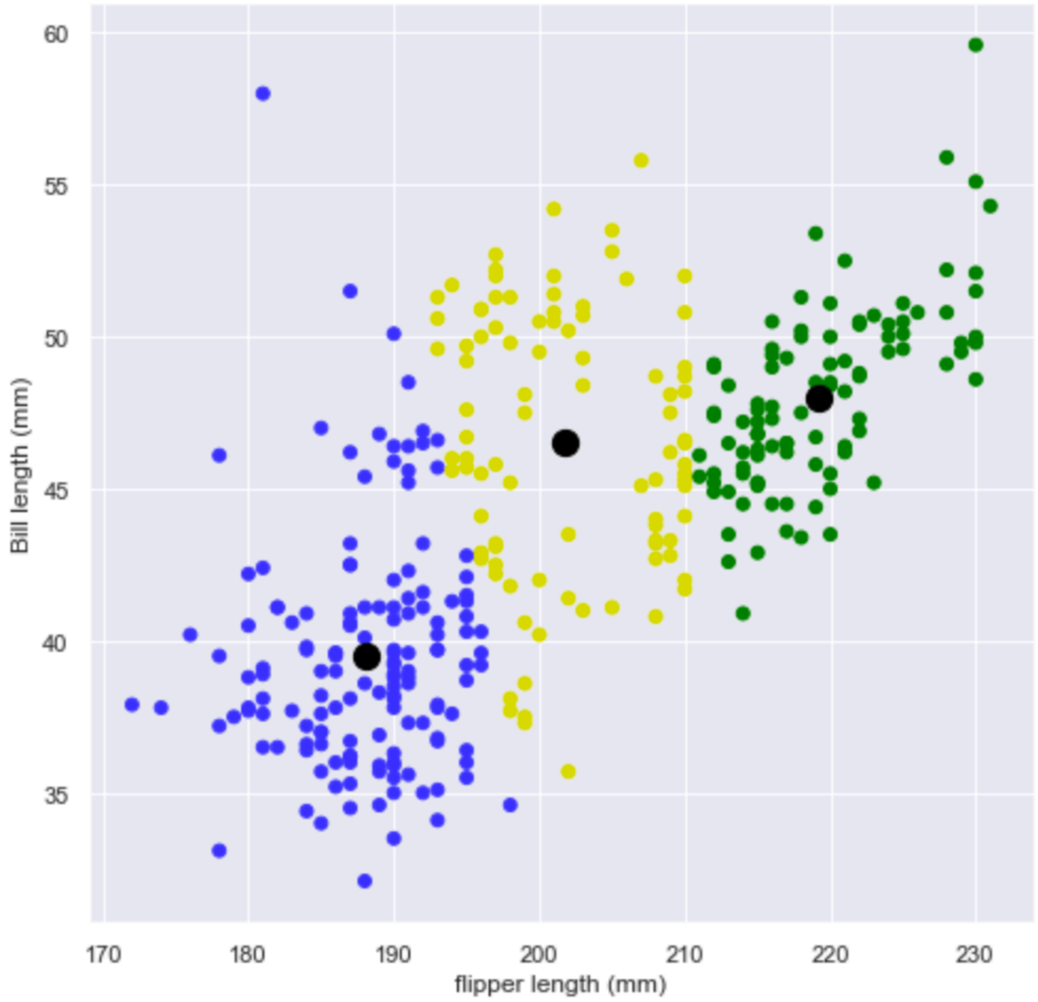
k-Means
Example

- Next assignment



k-Means
Example

- Final assignment



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