

# Projet d'Apprentissage Automatique

## Objectifs pédagogiques du projet

- Découvrir le B.A.BA de l'apprentissage automatique à travers l'expérience de la classification supervisée binaire
- Pédagogie particulière : réaliser par soi-même l'importance des concepts fondamentaux de l'apprentissage (mesure de performances, généralisation, importance de l'échantillon et notion de variance, notion d'hyper-paramètres, représentation vectorielle des données, etc.)



## La vraie vie

Que voit-on dans des images ?

- On observe: images de drones de qualités diverses
- Dans ces images, quelles sont les zones maritimes ? On le sait : étiquetage de chaque image = On y voit la mer (+1) ou pas (-1).
- L'ordi observe après coup une nouvelle image envoyée par le drone : est-ce une image qui comprend une zone maritime ? Quelle confiance dans cette précision ?

# Schématiquement

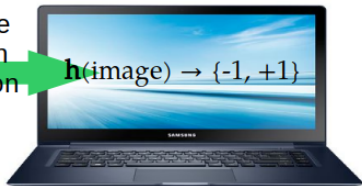
## Phase d'apprentissage

Échantillon étiqueté



Apprentissage  
d'une fonction  
de classification  
binaire  $h$

$$h(\text{image}) \rightarrow \{-1, +1\}$$



# Schématiquement

## Phase de décision (mise en production)



# Schématiquement

## Phase de décision (mise en production)



## Plus pragmatiquement

### Quelles observations ?

- Un échantillon d'images couleurs, de tailles et de résolutions différentes (répertoire `Data`)
- Une étiquette (catégorie) pour ces images : selon le sous-répertoire dans lequel elle se trouve
  - Sous-répertoire `Mer` = +1 = image maritime
  - Sous-répertoire `Ailleurs` = -1 = image non maritime

### Déterminer une fonction $f(\text{image}) \rightarrow \{\text{mer (+1), ailleurs (-1)}\}$

Valable pour les images déjà observées, mais aussi de façon générale face à n'importe quelle nouvelle image

### Remise du projet (pour évaluation)

- Un programme Python complet offrant une fonction de prédiction de la catégorie d'une nouvelle image passée en paramètre (coeff 2)
- Un rapport présentant l'architecture logicielle choisie, le(s) algorithme(s) d'apprentissage développés (coeff 1), les verrous rencontrés et les solutions apportées (entre 5 et 10 pages, pdf).
- La prédiction appliquée à une série de 40 images inconnues : les trois meilleures équipes gagnent un bonus.

# Organisation ce jour

## Constitution des groupes et brainstorming

- Trois à cinq personnes par groupe (tirage au sort des chefs temporaire d'équipe – ou sur candidature et vote – puis sélection des autres membres par désignation du chef d'équipe, ou par courte paille, aux dés, à Shifumi, etc.)
- Brainstorming de 20 minutes au sein de chaque équipe pour envisager les axes majeurs d'une première solution et les variantes possibles.

## Présentations de premières solutions et discussions

Pour chaque équipe :

- Présentation en 4 minutes de sa solution schématisée, possibilité de questions/réponses des autres équipes.
- Distribution des rôles, chaque membre se voit attribuer un (ou plusieurs) rôle(s) parmi : Chef, responsable scientifique, responsable technique, rapporteur, concepteur, testeur, développeur, relation client, etc.

## Le bon chemin (à prendre ou à laisser...)

### Quelques points clés

L'essentiel : revenir sur le cours du matin, et laisser aller son imagination. Se fixer assez vite une ébauche de solution.

- Importance d'un format homogène des données. Lequel ? Bibliothèques disponibles pour traiter les images ?
- On peut utiliser Naïves Bayes, mais n'y a-t-il pas d'autres algorithmes plus efficaces ? Peut-on créer son propre algorithme ?
- Peut-on jouer avec l'échantillon ? (l'augmenter par transformation des images existantes, ou par dédoublement de certaines, etc.) ?
- Peut-on modifier les images avant qu'elles ne soient en entrée du processus ? Les découper, enlever le bleu, les mettre en noir et blanc, etc. ?
- Comment organiser nos données et notre code pour travailler à plusieurs ? (ex. prototypes en C = interfaces de fonctions communicantes)
- Comment tester la qualité de ce que l'on fait ?