Chapitre 3
Comprendre et préparer les données

1. Introduction

Avant de commencer à créer nos propres algorithmes qui nous permettront d'obtenir des modèles performants, il y a une étape capitale que nous ne devrons jamais minimiser : **la préparation des données**.

En effet, tout projet d'IA nécessitera de récolter des données, de les explorer en les visualisant pour mieux les comprendre, de les nettoyer et de les rendre disponibles pour pouvoir créer des modèles adaptés et performants. En fonction de celles-ci et des résultats souhaités, nous serons amenés à choisir certains algorithmes plus adaptés que d'autres. La qualité des données est la clé de la réussite de tout projet d'IA. Sans des données propres, pertinentes et bien structurées, même les algorithmes les plus sophistiqués échoueront.

Partons à la découverte de cette étape fondamentale en parcourant les thèmes suivants :

- types de données ;
- types d'apprentissages ;
- solutions de visualisation des données ;
- récolte, nettoyage et imputation des données ;
- exploration et analyse des données appliquées à PHP ;

– prétraitement des données ;
– solutions de réduction des dimensions.

À la fin de ce chapitre, nous connaîtrons les différents types de données qu'il sera possible de rencontrer à la suite de leur récolte, nous aurons à notre disposition des outils pour les visualiser, les nettoyer puis les analyser afin de pouvoir mieux les comprendre et les préparer pour une exploitation.

2. Types de données

Dans un énoncé de problème à traiter avec l'IA, la première étape est de récolter les données qui nous permettront d'éduquer et d'entraîner nos modèles qui auront pour vocation de solutionner notre problématique. Dans les ensembles que nous aurons, il sera possible de rencontrer une large variété de types de données. Voici une liste des types les plus courants.

2.1 Données numériques

Les données numériques peuvent être **continues** ou **discrètes**. Elles sont **discrètes** si elles ne peuvent prendre qu'une quantité limitée de valeurs dans un intervalle donné (exemple : l'effectif présent dans une salle ne peut prendre que les valeurs 0, 1, 2... 30). Elles sont **continues** si elles peuvent prendre une infinité de valeurs dans un intervalle défini (exemple : la température d'une pièce à vivre).

2.2 Données catégorielles

Les données **catégorielles** prennent des valeurs qui représentent des catégories distinctes dans un panel donné.

Parmi ces données catégorielles, nous pourrions par exemple trouver les types de produits (électronique, vêtement, alimentation), les genres (masculin, féminin, autre) ou même les couleurs (rouge, bleu, vert).

2.3 Données ordinales

Les données **ordinales** sont des données catégorielles avec un ordre intrinsèque.

Par exemple, des données comme les niveaux d'éducation (primaire, secondaire, universitaire), les échelles de satisfaction (insatisfait, neutre, satisfait) ou les classements de performances (mauvais, moyen, bon) sont ordinales.

2.4 Données textuelles

Les données **textuelles** sont des données sous forme de texte libre.

Par exemple, des commentaires clients, des articles de blog ou le corps d'un courrier électronique sont des données textuelles.

2.5 Données temporelles

Les données **temporelles** sont des données qui sont indexées dans le temps.

Les séries chronologiques de ventes quotidiennes, les données de capteurs collectées chaque minute ou les enregistrements des températures horaires sont des données temporelles que nous pourrions retrouver dans nos projets.

2.6 Données géospatiales

Les données **géospatiales** sont des données associées à des coordonnées géographiques. Par exemple, la localisation GPS de véhicules, les coordonnées de magasins et les densités de population font partie de ce type de données.

2.7 Données multimédias

Les données **multimédias** peuvent être des données audio, d'images ou de vidéo.

Les enregistrements de voix pour des assistants vocaux, les sons d'animaux pour des projets de reconnaissance de faune, les vidéos de sports pour l'analyse de performance, les radiographies médicales, les images satellites et les photos de produits pour des sites de commerce électronique sont de bons exemples de telles données.

2.8 Données logiques/binaires

Les données **binaires** sont des données représentant deux états (vrai/faux, oui/non), comme les indicateurs d'activation de fonctionnalités (activé/désactivé), les réponses à des questions oui/non ou la présence/absence de caractéristiques spécifiques.

Ces différents types de données peuvent être utilisés seuls ou combinés pour résoudre des problèmes variés en IA, chaque type de données ayant ses propres méthodes et traitements. En effet, nous allons apprendre comment appréhender ces données afin qu'un algorithme d'IA puisse les utiliser. Car n'oublions pas qu'une machine n'a pas intrinsèquement la capacité de comprendre la symbolique d'une donnée, il faudra donc transformer les données numériquement en conservant leurs principales significations et caractéristiques. Avant d'analyser les données, il est important de situer notre problème pour identifier le mode d'apprentissage que nous allons devoir utiliser pour résoudre notre besoin.

3. Types d'apprentissage

En Machine Learning, nous rencontrerons principalement deux approches pour éduquer et entraîner nos modèles :

- **L'apprentissage supervisé** : ce type d'apprentissage utilise des données étiquetées (souvent le résultat attendu), où chaque exemple d'entraînement est associé à une réponse ou à une sortie connue.
  - **Si les étiquettes sont des données numériques continues**, nous sommes dans le cadre d'un problème de **régression**.
  - **Si les étiquettes sont des données catégorielles**, nous sommes dans le cadre d'un problème de **classification**.

Illustration concrète

Vous souhaitez apprendre les noms des capitales de chaque pays du monde à votre enfant. Vous lui posez des questions, vous attendez qu'il vous propose une solution. Ensuite, vous lui donnez la capitale espérée. Ainsi, il corrige son comportement en fonction de la réponse (correction ou confirmation).

- **L'apprentissage non supervisé** : ce type d'apprentissage utilise des données non étiquetées. Le modèle cherche des structures ou des patterns inhérents dans les données sans supervision humaine.

Illustration concrète

Vous avez un sac avec de multiples objets, vous devez les classer en plusieurs groupes. À vous de chercher des critères pour définir ces groupes et les créer. Certains les classeront par types d'utilisation, d'autres par couleurs, etc. Cela dépendra de l'algorithme mis en œuvre.

Ainsi, en fonction de l'ensemble de données qui nous sera fourni, nous pourrons déjà définir notre mode d'apprentissage. Cela nous permettra par la suite d'identifier les algorithmes qui permettront de résoudre notre problème. En effet, nous aurons tout un arsenal d'algorithmes pour effectuer ces apprentissages et certains conviendront mieux que d'autres à certains types de données et aussi en fonction de leur nombre.

En visualisant et en analysant les données plus en détail, nous pourrons ensuite préciser nos choix d'algorithmes.

4. Solutions de visualisation des données

Pour bien cerner notre problème et mieux l'envisager par la suite, observons nos données pour mieux les comprendre. L'objectif de cette phase d'analyse est d'avoir une estimation de la distribution des données pour se faire une idée de leur représentativité. Les données sont-elles bien réparties ou notre échantillon ne représente-t-il que certains types ou intervalles en particulier ? Cela nous permettra ensuite de connaître les limites potentielles de notre modèle.

PHP n'étant pas un langage à vocation de création visuelle, nous allons utiliser une librairie JavaScript optimisée pour la visualisation des données. Notre choix se portera sur **Plotly.js**, une surcouche du célèbre **D3.js**.

Afin d'éviter d'y passer trop de temps et ainsi nous permettre de nous concentrer sur nos besoins réels, utilisons une librairie PHP permettant la génération du code pour les diagrammes directement depuis PHP en quelques lignes simples disponibles à cette adresse :
https://github.com/LouisAUTHIE/Php2Plotly/

Le plus simple sera de déléguer son installation à **Composer** via la commande suivante :

```
composer require louisauthie/php2plotly
```

La seule nécessité en amont de son utilisation est de nous occuper de l'inclusion de la librairie Plotly.js dans notre code HTML, à l'intérieur de la balise `<head>` de notre page. Par exemple, nous pourrions utiliser le code suivant :

```
<script src="js/plotly-2.32.0.min.js" charset="utf-8"></script>
```

Présentons rapidement les types de diagrammes à notre disposition. Nous en profiterons pour proposer un exemple de mise en œuvre pour chacun des diagrammes.

4.1 Diagramme à bâtons

Ce type de diagramme permet de visualiser les effectifs en ordonnées en fonction de catégories en abscisses. En voici un ci-après, illustrant les effectifs de personnes par tranches d'âge dans une population définie.

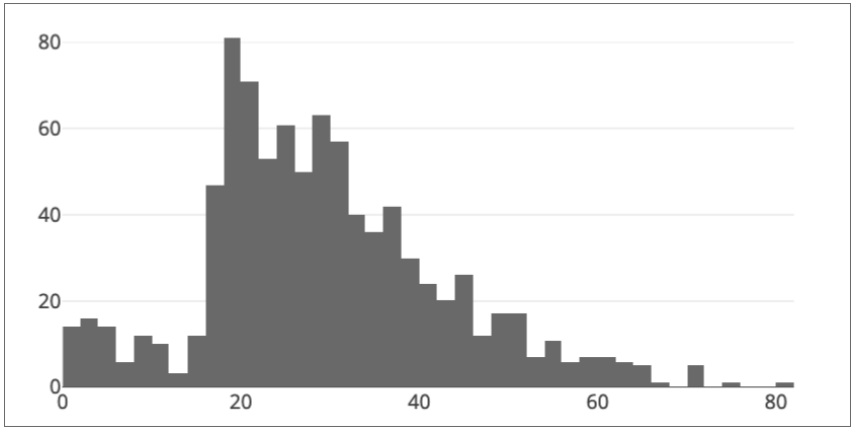

Figure 3.1 - exemple de diagramme à bâtons

Ce type de diagramme sera très utilisé dans nos études de données pour les données numériques discrètes (sans retraitement nécessaire) et numériques continues (avec une création d'intervalles de données pour définir des catégories en abscisses).

En utilisant notre librairie de visualisation depuis PHP, nous pouvons inclure un diagramme à bâtons avec le code suivant :

```
<div id="bar" style="width:600px;height:400px;"></div>

<?php
       $bar = new BarChart('bar', ['x' => ["Football", "Rugby",
"Handball", "BasketBall"], 'y' => [34, 15, 18, 26]]);
       echo '<script>'.$bar->render().'</script>';
?>
```

Voici le rendu de cet exemple :

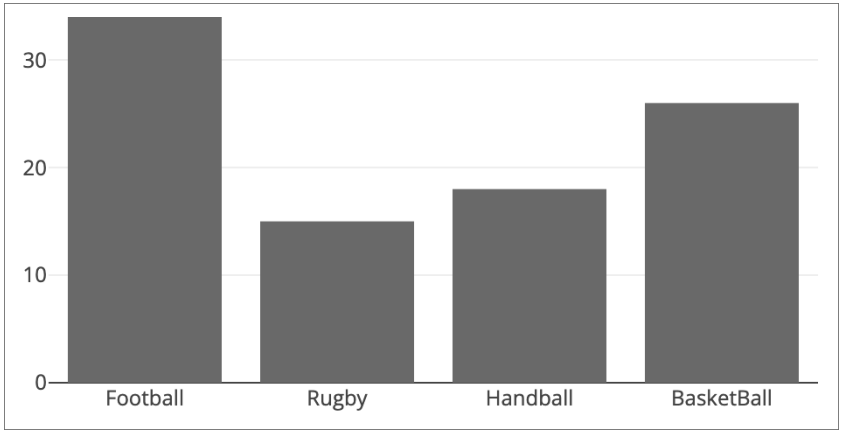

Figure 3.2 – rendu de notre exemple de diagramme à bâtons

Nous instancions simplement l'objet `BarChart` dont le constructeur prend en arguments l'id du conteneur dans lequel afficher le diagramme, et un tableau associatif avec la clé `x` pour les abscisses et la clé `y` pour les ordonnées.

4.2 Diagramme en camembert

Ce type de diagramme permet de visualiser les proportions des effectifs de chaque catégorie au sein de toutes les autres catégories d'un même ensemble de données.

En voici un exemple simple présentant les proportions des effectifs de personnes de sexes masculin et féminin dans la population étudiée.

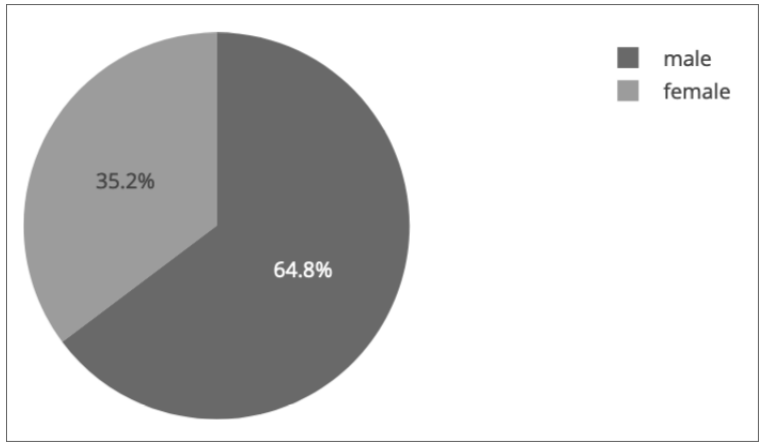

Figure 3.3 - exemple de diagramme en camembert

Ce type de diagramme est particulièrement approprié à la visualisation des données catégorielles, ordinales et binaires.

Un exemple de code pour créer un tel diagramme pourrait être le suivant :

```
<div id="pie" style="width:600px;height:400px;"></div>
<?php
   $pie = new PieChart('pie', ['values' => [10, 15, 13, 17],
'labels' => ["Chats", "Chiens", "Oiseaux", "Oursons"]],
['height' => 400, 'width' => 600]);
   echo '<script>'.$pie->render().'</script>';
?>
```