k – Plus proches voisins

Chargement des librairies

Entrée [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
```

1. Plus proches voisins pour la régression

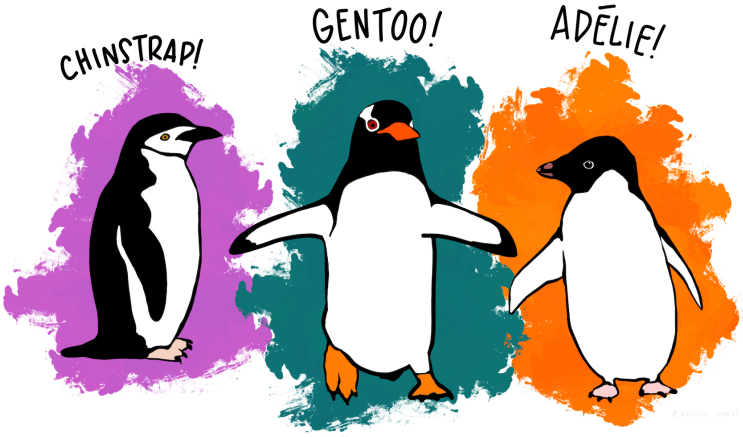
Chargement des données:

Dans cette partie pratique nous allons utiliser un jeu de données décrivant des manchots, le jeu Palmer Penguins (https://allisonhorst.github.io/palmerpenguins/).

Le jeu de données contient des informations décrivant un certain nombre de manchots appartenant à trois espèces :

- Manchot d'Adélie (*Pygoscelis adeliae*), `Adelie` dans les données
- Manchot papou (*Pygoscelis papua*), `Gentoo` dans les données
- Manchot à jugulaire (*Pygoscelis antarcticus*), `Chinstrap` dans les données.

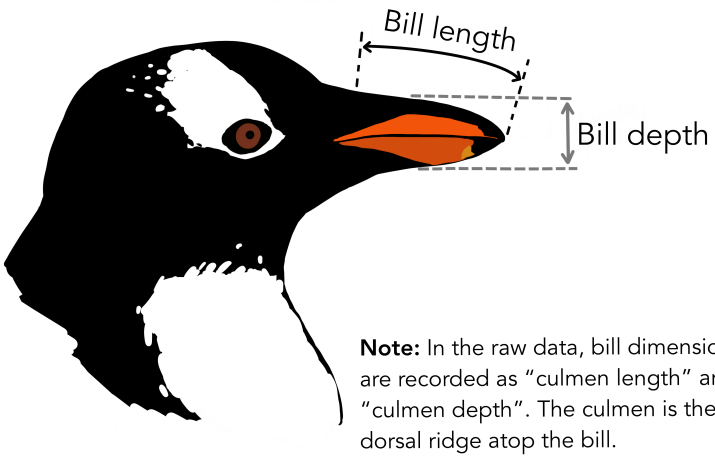
Les trois espèces de manchots:



Pour chacun des manchots, le jeu de données contient

- la longueur de son bec en mm (`bill_length_mm`)
- la hauteur de son bec en mm (`bill_depth_mm`)
- la longueur de ses palettes natatoires en mm (`flipper_length_mm`)
- son poids en g (`body_mass_g`)

Caractéristiques du bec:



Entrée [15]:

```
penguins = pd.read_csv("data/penguins.csv")
penguins.head()
```

Out[15]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	se
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	ma
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	fema
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	fema
3	Adelie	Torgersen	36.7	19.3	193.0	3450.0	fema
4	Adelie	Torgersen	39.3	20.6	190.0	3650.0	ma

Entrée [17]:

```
X = penguins[["bill_length_mm", "bill_depth_mm", "flipper_length_mm"]].to_numpy()
y = penguins["body_mass_g"]
```

Séparation des données en jeu d'entraînement et jeu de test

Entrée [18]:

```
from sklearn.model_selection import train_test_split
(X_train, X_test, y_train, y_test) = train_test_split(X, y, test_size=0.3, random_state=2
```

Les algorithmes de plus proches voisins sont implémentés dans le module `neighbors` (https://scikit-learn.org/stable/modules/classes.html?highlight=neighbors#module-sklearn.neighbors) de `scikit-learn` . Pour la régression, nous utilisons la classe `KNeighborsRegressor` (https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html).

Nous allons fixer ici le nombre de plus proches voisins k à 7.

Entrée [9]:

```
from sklearn import neighbors
```

Nous suivons toujours les étapes habituelles :

1. Instancions un objet de la classe `KNeighborsRegressor` .

Entrée [12]:

```
knnreg = neighbors.KNeighborsRegressor(n_neighbors=7)
```

2. Entraînons cet objet sur les données d'entraînement avec la méthode `fit` :

Ici, l'entraînement consiste uniquement à définir l'ensemble des observations étiquetées parmi lesquelles chercher les voisins d'une observation.

Entrée [20]:

```
knnreg.fit(X_train, y_train)
```

Out[20]:

```
KNeighborsRegressor(n_neighbors=7)
```

3. Enfin, prédisons les étiquettes des données du jeu de test en utilisant la méthode `predict` :

Entrée [21]:

```
y_test_pred = knnreg.predict(X_test)
```

Performance du modèle

Entrée [22]:

```
from sklearn import metrics

print("La RMSE de notre modèle est %.2f g" % (metrics.mean_squared_error(y_test, y_test_p
print("Le coefficient de détermination de notre modèle est R2 = %.2f" % (metrics.r2_score
```

```
La RMSE de notre modèle est 388.20 g
Le coefficient de détermination de notre modèle est R2 = 0.77
```

Entrée [ ]:

```

```