0.1 Introduction

L'algorithme des k plus proches voisins appartient à la famille des algorithmes d'apprentissage automatique (machine learning).
L'idée d'apprentissage automatique ne date pas d'hier,puisque le terme de machine learning a été utilisé pour la première fois par l'informaticien américain Arthur samuel en 1959.Les algorithmes d'apprentissage automatique ont connu un fort eegain d'intérêt au début des années 2000 notamment grâce à la quantité de données disponibles sur Internet.
L'algorithme des k-plus proches voisins est un **algorithme d'apprentissage supervisé** (...)

0.2 Principe de l'algorithme

L'algorithme des k plus proches voisins **ne nécessite pas de phase d'apprentissage** à proprement parler: il faut juste stocker le jeu de données d'apprentissage.

Soit un ensemble E contenant n données labellisées : $E = \{(y_i, \overrightarrow{x_i})\}$ avec i compris entre 1 et n , où y_i correspond à la classe (le label) de la donnée i et où le vecteur $\overrightarrow{x_i}(x_{1i}, x_{2i}, ..., x_{pi})$ de dimension p représente les variables prédictrices de la donnée i.Soit une donnée u qui n'appartient pas à E et qui ne possède pas de label (u est uniquement caractérisée par un vecteur $\overrightarrow{x}_u$ de dimension p).

Soit d une fonction qui renvoie la distance entre la donnée u et une donnée quelconque appartenant à E.
Soit un entier k inférieur ou égal à n.Voici le principe de l'algorithme des k plus proches voisins:

1. On calcule les distances entre la donnée u et chaque donnée appartenant à E à l'aide de la fonction d.
2. On retient les k données du jeu de données E les plus proches de u.
3. On attribue à u la classe qui est la plus fréquente parmi les k données les plus proches.

0.3 Etude d'un exemple

0.3.1 les données

Nous avons choisi de nous baser sur le jeu de données "iris de Fisher" .
Dans ce jeu de données nous avons pour chaque entrée :

1. La longueur des sépales (en cm)
2. La largeur des sépales (en cm)
3. La longueur des pétales (en cm)
4. La largeur des pétales (en cm) L'espèce d'iris : Iris setosa , iris virginica ou iris versicolor(label)

Il est possible de télécharger ces données au format csv par exemple sur le site GitHub Gist.

Une fois ces données téléchargées,il est nécessaire de les modifier à l'aide d'un tableur:

- Dans un souci de simplification, nous avons choisi de travailler uniquement sur la taille des pétales.Nous avons donc supprimer les colonnes "sepal_length" et "sepal_width"
- Il est nécessaire d'encoder les espèces avec des chiffres :
  - 0 pour Iris setoa
  - 1 pour Iris virginica
  - 2 pour Iris versicolor (ce processus d'encodage des données textuelles est relativement classique en apprentissage automatique).

0.3.2 Bibliothèques Python utilisées

Nous allons utiliser 3 bibliothèques Python:

- **Pandas** : qui va nous permettre d'importer les données issues du fichier csv.
- **Matplotlib** qui va nous permettre de visualiser les données.
- **Scikit-learn** qui propose une implémentation de l'algorithme des k plus proches voisins.

0.3.3 Première visualisation des données

Une fois le fichier csv modifié , il est possible d'écrire un programme permettant de visualiser les données sous forme de graphique (abscisse :"petal_length",ordonnée:"petal_width"):

```
1  import pandas
2  import matplotlib.pyplot as plt
3  iris=pandas.read_csv("iris.csv")
4  x=iris.loc[:,"petal_length")]
5  y=iris.loc[:,"petal_width"]
6  lab=iris.loc[:,"species"]
7  plt.scatter(x[lab==0],y[lab==0],color='green',label='setoa')
8  plt.scatter(x[lab==1],y[lab==1],color='red',label='versicolor')
9  plt.scatter(x[lab==2],y[lab==2],color='blue',label='virginica')
10 plt.legend()
11 plt.show()
```

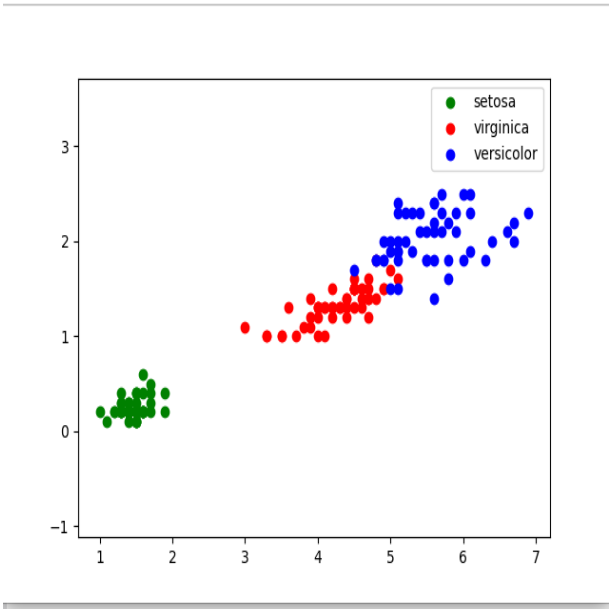

Figure 1: représentation des données

0.3.4 utilisation de l'algorithme des k plus proches voisins

1. On ajoute une donnée anonymisée:

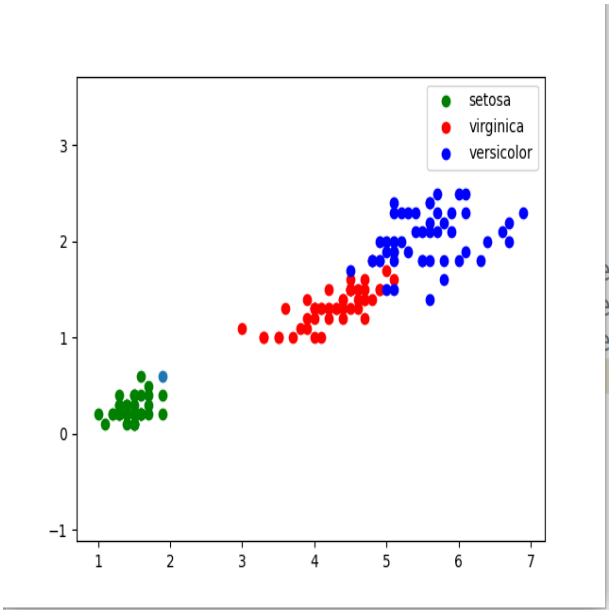

Figure 2: Ajout d'une donnée non labellisée

A quelle espèce appartient-elle?
Quelle stratégie utiliser pour répondre à la question?

2. Un cas plus difficile:
Répondre à la même question avec la nouvelle donnée ajoutée ci-dessous.

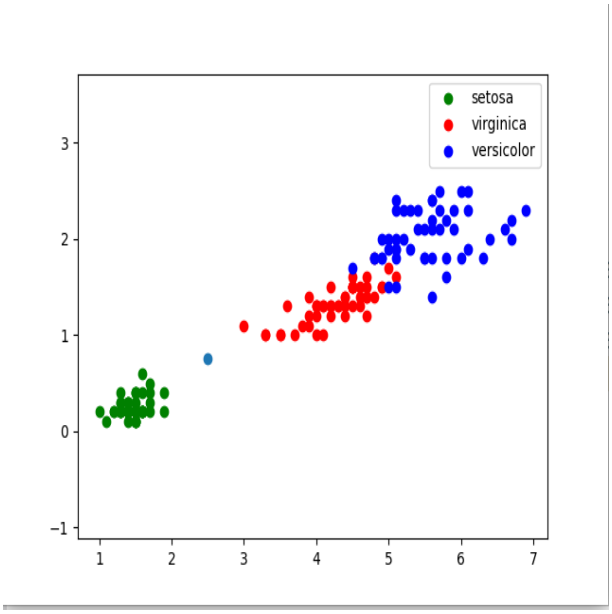

Figure 3: Ajout d'une donnée non labellisée

3. stratégie possible :

- on calcule la distance entre notre point (largeur du pétale = 0,75 cm ; longueur du pétale = 2,5 cm) et chaque point issu du jeu de données "iris" (à chaque fois c'est un calcul de distance entre 2 points) ;
- on sélectionne uniquement les k distances les plus petites (les k plus proches voisins) ;
- parmi les k plus proches voisins, on détermine quelle est l'espèce majoritaire. On associe à notre "iris mystère" cette "espèce majoritaire parmi les k plus proches voisins". Dans l'exemple évoqué ci-dessus (largeur pétale = 0,75 cm ; longueur pétale = 2,5 cm), pour k=3, nous obtenons graphiquement :

Conclusion:Un iris ayant une largeur de pétale égale à 0,75 cm et une longueur de pétale égale à 2,5 cm a une "forte" probabilité d'appartenir à l'espèce setosa.

0.3.5 Utilisation de scikit-learn

La bibliothèque scikit-learn propose un grand nombre d'algorithmes liés à l'apprentissage automatique (c'est sans aucun doute la bibliothèque la plus utilisée en apprentissage automatique).Parmi tous ces algorithmes , scitit-learn propose l'algorithme des k plus proches voisins.Voici un programme Python permettant de résoudre le problème évoqué ci-dessus(largeur pétale = 0,75 cm; longueur pétale=2,5cm):

```
import pandas
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier

#traitement CSV
iris=pandas.read_csv("iris.csv")
x=iris.loc[:,"petal_length")]
y=iris.loc[:,"petal_width"]
lab=iris.loc[:,"species"]
#fin traitement CSV

#valeurs
longueur =2.5
largeur =0.75
k=3
#fin valeurs

#graphique
plt.scatter(x[lab==0],y[lab==0],color='green',label='setoa')
plt.scatter(x[lab==1],y[lab==1],color='red',label='virginica')
plt.scatter(x[lab==2],y[lab==2],color='blue',label='versicolor')
plt.legend()
#fin graphique

#algo knn
d=list(zip(x,y))
model=KNeighborClassifier(n_neighbors=k)
model.fit(d,lab)
prediction=model.predict([longueur,largeur]])
#fin algo knn

#affichage résultats
txt="Résultat :"
if prediction[0]==0:
    txt=txt + "setosa"
if prediction[0]==1:
    txt=txt +"virgica"
if prediction[0]==2:
    txt=txt +"versicolor"
plt.text(3,0.5,f"largeur : {largeur} cm longueur : {longueur} cm", fontsize=12)
plt.text(3,0.3,f"k : {k}" , fontsize=12)
plt.text(3,0.1, txt, fontsize=12)
#fin affichage résultats

plt.show()
```

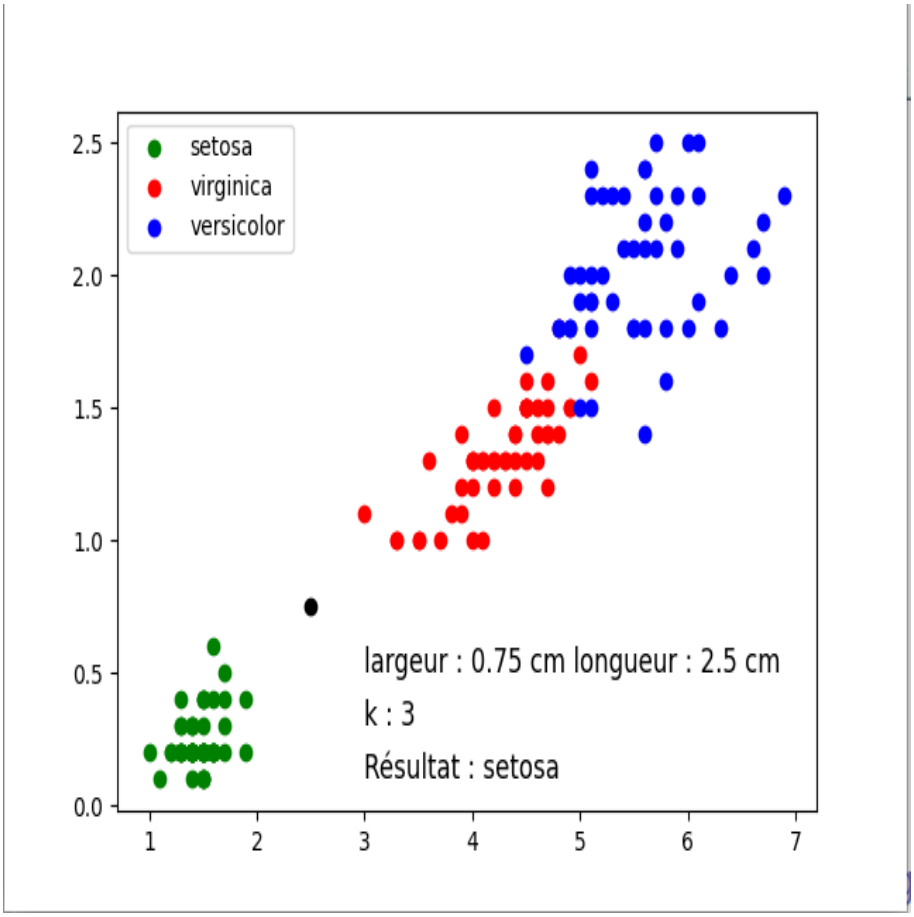

Figure 4: – 3 plus proches voisins à l'aide de scikit-learn dans le cas : largeur pétale = 0,75 cm ; longueur pétale = 2,5 cm

1. Modifier le programme ci-dessus afin d'étudier les changements induits par la modification du paramètre k (notamment pour k=5) en gardant toujours les mêmes valeurs de largeur et de longueur (largeur pétale = 0,75 cm ; longueur pétale = 2,5 cm).
2. travailler avec d'autres valeurs de longueur et largeur.